Uncertainty-aware Interactive Imitation Learning for Robot Manipulation

Franzese, G.

DOI
10.4233/uuid:f2fc909c-49bf-46ed-880a-32743f1a6a8e

Publication date
2024

Document Version
Final published version

Citation (APA)
Franzese, G. (2024). *Uncertainty-aware Interactive Imitation Learning for Robot Manipulation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:f2fc909c-49bf-46ed-880a-32743f1a6a8e

Uncertainty-aware Interactive Imitation Learning for Robot Manipulation

Giovanni Franzese

Uncertainty-Aware Interactive Imitation Learning for Robot Manipulation

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Monday the 4th of November 2024 at 15:00 o'clock

by

Giovanni FRANZESE

Master of Science in Mechanical Engineering, Politecnico di Milano, Italy,
born in San Giuseppe Vesuviano, Italy

This dissertation has been approved by the promotors.

Rector Magnificus,	Chairperson
Dr.-Ing. J. Kober,	Delft University of Technology, *promotor*
Dr. L. Peternel,	Delft University of Technology *copromotor*

Composition of the doctoral committee:

Prof. dr. R. Babuška,	Delft University of Technology
Prof. dr. B. Nemec,	Jozef Stefan Institute, Slovenia
Dr. M. Kok,	Delft University of Technology
Dr. A. Saccon,	Eindhoven University of Technology
Prof. dr. ir. D. A. Abbink,	Delft University of Technology, *reserve member*

Funded by the European Union

European Research Council
Established by the European Commission

Keywords: Gaussian Process, Human-in-the-loop Robot Manipulation, Learning from Demonstrations, Task-Parameterized Robot Learning

Printed by: Gildeprint

Cover: The School of Delft: where robots learn from humans. Inspired by The School of Athens by Raphael. Created by Giovanni Franzese and image generation tools.

ISBN 978-94-6366-946-7

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

To my dad, Michele, for teaching me resilience;
to my lovely mother, Geni, for shaping my critical reasoning;
and to my brother, Vittorio, for making every day of my life better.

Contents

SUMMARY

While Artificial Intelligence (AI) is geared towards automating tasks like writing and designing, the challenge persists in finding adequate human resources for tasks such as handling luggage in and out of airplanes or harvesting produce in greenhouses. Nonetheless, the demand to tailor robotic abilities to diverse scenarios, ranging from agriculture to household chores, necessitates a general-purpose morphology for the robot, such as a dexterous arm, along with sufficient sensory capabilities and intelligence to swiftly adjust to new situations. Despite the prevalence of click-baiting videos shared online, current robot technologies have yet to address this requirement adequately.

The primary obstacle hindering robot manipulators from effectively performing daily chores, aiding in supermarkets, and harvesting fruits from fields is the insufficient data available to construct a robust model of the world. Typically, autonomously exploring their surroundings and determining optimal strategies is considered unsafe and impractical. A more effective approach to imparting knowledge to robots involves human supervision. Ideally, this entails interactive supervision where robots can seek clarification when uncertain about a situation, and humans can intervene when the robot's actions are incorrect or fail to meet the required performance. Moreover, when receiving instructions or asking for them, the robot should quantify the confidence in the interpretation of the corrections. This thesis makes significant contributions to the field of interactive robot learning by introducing various uncertainty-aware methods. These methods facilitate enhancements in data efficiency during learning and safety during execution.

Before delving into the main contributions, Chapter 2 introduces the reader to the topic of Interactive Imitation Learning (IIL) and the different modalities that can be used to give feedback, from evaluative to corrective, underlying the importance of uncertainty quantification on the robot belief. For this reason, Chapter 3, introduces the foundations of the main function approximator used in this thesis, i.e. Gaussian Process (GP), to learn behaviors while quantifying uncertainties. The chapter highlights how a GP is trained given the evidence of the data and the corrections and how predictions of the mean and the variance of the actions are obtained. Particular attention is given to how GP models can be used for efficient updating and aggregation of online data and how to analytically estimate the uncertainty rate of change.

The proposed function approximator is first applied in Chapter 4. The presented machine learning framework allows the robot to learn complex manipulation tasks from interactive demonstrations. Essentially, the user needs to show a kinesthetic demonstration to the robot, i.e. dragging the robot around in a fully compliant modality to transfer their knowledge on a desired skill, e.g. cleaning a table or inserting a plug in a socket. The experiments highlight how the quantification and the rejection of uncertainties can be used to bring the robot always close to high-confidence regions. Moreover, the GP online model update is used to aggregate the corrections received from the user to reshape the learned attractor and the stiffness field. This ensures that the proper force is executed in the correct direction for instance when cleaning a table.

To extend the learning of a skill to the whole robot pose and gripper, Chapter 5 studies how to address this with GP and with the least amount of demonstrations and corrections. Moreover, the experiments focus on teaching human-like skills to robots by exploiting the possibility of giving incremental corrections. In particular, novice users, are asked to perform the picking task of objects in one fluid motion by teaching the complete pose and gripper behavior. The execution of the skill without any supervision is usually too slow or knocks the object down before closing the gripper. Nevertheless, after providing feedback, novice users were able to incrementally shape the robot's velocity to perform the picking at non-zero velocity, without knocking the object and correcting for any delay in gripper dynamics.

However, learning skills only relying on the current robot's Cartesian position can be a limitation since it cannot encode skills that entail overlapping, e.g. when approaching a goal and then moving back on the same trajectory. This motivates Chapter 6 which formulates a new trajectory encoding to teach single or bimanual manipulation skills while being safe around humans with constrained velocity and force actuation. The user study also investigates the effectiveness of giving kinesthetic corrections, i.e. by simply touching the robot, and validating this in teaching bimanual skills. Teaching two manipulators at the same time or correcting them using teleoperation devices can become overwhelming. Hence, the method explores adjusting movements interactively through kinesthetic perturbations rather than re-teaching skills entirely from scratch due to imprecise attempts.

Despite the successful applications of the proposed methods in single and bimanual motion skills, during task learning, the robot must not only master the motor aspect but also be attentive to the context, such as the object's location or shape. This motivates Chapter 7, which emphasizes the generalization of acquired motor skills across various contexts. The proposed approach hinges on GP theory to acquire a non-linear transformation map from the demonstrated task space to the execution space while preserving and propagating uncertainties. Through experiments involving tasks such as pick-and-place operations, dressing human arms, and cleaning surfaces, it is demonstrated how the robot can generalize the execution by transforming the attractor, orientation, and stiffness policy to numerous new scenario configurations even with just a single demonstration of the skill.

In Chapter 8, the concept of task parametrization and uncertainty awareness is expanded to over-parameterizing the context, such as by tracking more objects than required. The proposed algorithm would prompt user attention when encountering ambiguity, like when multiple detected objects could be the goal of the skill. Decision ambiguity can be resolved by various feedback modalities, such as pushing the robot, moving it, or providing reward/punishment. A user study also highlighted the preference of novice users for not giving conventional kinesthetic demonstrations but only intervening when necessary.

Samenvatting

Terwijl kunstmatige intelligentie (AI) gericht is op het automatiseren van taken zoals schrijven en ontwerpen, blijft het een uitdaging om geschikte menselijke arbeidskrachten te vinden voor taken zoals het laden en lossen van vliegtuigbagage of het oogsten van landbouwproducten in kassen. Desondanks is er de vraag om de vaardigheden van robots aan te passen aan diverse scenario's, variërend van landbouw tot huishoudelijke taken. Dit vereist dat robots een universele morfologie hebben, met voldoende sensorisch vermogen en intelligentie om zich snel aan te kunnen passen aan nieuwe situaties. Ondanks de verspreiding van sensationele video's online, is de huidige robottechnologie nog niet in staat om effectief aan deze eisen te voldoen.

Het belangrijkste obstakel dat robots ervan weerhoudt om effectief dagelijkse taken uit te voeren, hulp te bieden in supermarkten of fruit te oogsten in het veld, is het gebrek aan benodigde data om betrouwbare modellen van de echte wereld te construeren. Gewoonlijk wordt het als onveilig of onpraktisch beschouwd om robots zelfstandig te laten verkennen en de beste strategieën te laten bepalen. Menselijk toezicht maakt het effectiever om robots kennis bij te brengen. Idealiter is dit interactief toezicht, waarbij robots om verduidelijking kunnen vragen wanneer ze onzeker zijn en mensen kunnen ingrijpen wanneer de acties van de robots onjuist zijn of niet voldoen aan de vereiste prestatie. Bovendien moet de robot in staat zijn om zijn vertrouwen in de interpretatie van de correcties te kwantificeren bij het krijgen van of vragen naar instructies. Deze thesis levert een significante bijdrage aan het vakgebied van interactief robotleren door verschillende methoden te introduceren die rekening houden met onzekerheid. De methoden faciliteren verbetering van zowel de data-efficiëntie tijdens het leren als de veiligheid tijdens de uitvoering.

Hoofdstuk 2 introduceert de lezer tot het onderwerp Interactive Imitation Learning (IIL) en de verschillende vormen waarop feedback kan worden gegeven, van evaluatief tot correctief, wat ten grondslag ligt aan het belang van het kwantificeren van de onzekerheid over dat wat de robot gelooft. Daaropvolgend introduceert Hoofdstuk 3 de grondbeginselen van Gaussiaanse Processen (GP) voor het leren van gedrag terwijl tegelijkertijd de onzekerheden gekwantificeerd worden. Dit hoofdstuk benadrukt hoe een GP getraind wordt gegeven het bewijsmateriaal uit de data en de correcties en hoe voorspellingen van zowel het gemiddelde als de variantie van acties worden verkregen. Specifieke aandacht wordt besteed aan hoe GP-modellen gebruikt kunnen worden voor het efficiënt updaten en samenvoegen van online data en hoe de mate van verandering van de onzekerheid analytisch benaderd kan worden.

Hoofdstuk 4 past deze methode toe zodat een robot complexe manipulatietaken kan leren door middel van interactieve demonstraties. De gebruiker geeft de robot een kinesthetische demonstratie, d.w.z. een volledig meegaande robot wordt fysiek rondbewogen om kennis van een specifieke vaardigheid over te brengen, zoals het schoonmaken van een tafel of het inpluggen van een stekker. De experimenten laten zien hoe de kwantificatie en het verwerpen van onzekerheden gebruikt kan worden om de robot zich altijd te laten bevinden in regio's met een hoog vertrouwen. Bovendien wordt de GP online model-update gebruikt om de

geleerde attractor en het stijfheidsveld aan te passen aan de hand van de correcties die door de gebruiker worden gegeven. Dit zorgt ervoor dat de juiste kracht in de juiste richting wordt uitgeoefend, bijvoorbeeld tijdens het schoonmaken van een tafel.

Hoofdstuk 5 bestudeert hoe de manier van leren van een vaardigheid uitgebreid kan worden naar een algehele robothouding en grijper met zo min mogelijk demonstraties en correcties. Daarnaast focussen de experimenten op het aanleren van mensachtige vaardigheden aan robots door gebruik te maken van de mogelijkheid om interactieve correcties te geven. Specifiek beginnende gebruikers wordt gevraagd om de taak van het oppakken van objecten in één vloeiende beweging uit te voeren en daarbij de volledige robothouding aan te leren en hoe te grijpen. Het uitvoeren van de vaardigheid zonder enig toezicht is over het algemeen te langzaam of het object wordt omgestoten voordat het gegrepen wordt. Echter, na feedback te geven waren de beginnende gebruikers in staat om incrementeel de snelheid van de robot zo vorm te geven dat deze het object met een snelheid groter dan nul kon oppakken zonder het omver te stoten en met compensatie voor vertraging in de dynamica van de grijper.

Het uitsluitend vertrouwen op de huidige Cartesische positie van de robot heeft de beperking dat geen vaardigheden geleerd kunnen worden met overlappende posities, bijvoorbeeld wanneer na het bereiken van een doel via dezelfde weg terug wordt bewogen. Dit motiveert Hoofdstuk 6, waarin een nieuwe koerscodering wordt geformuleerd voor het aanleren van enkelvoudige en tweehandige taken, op een manier die veilig is rond mensen door de begrensde kracht- en snelheidsaansturing. Het gebruikersonderzoek onderzoekt ook de effectiviteit van het geven van kinesthetische correcties, d.w.z. door de robot aan te raken, voor bimanuele vaardigheden. Twee robotarmen tegelijkertijd onderwijzen of corrigeren d.m.v teleoperatie kan overweldigend zijn. In plaats daarvan onderzoekt de methode het interactief aanpassen van bewegingen d.m.v kinesthetische perturbaties.

Ondanks de succesvolle toepassing van de voorgestelde methoden voor enkelvoudige en tweehandige bewegingsvaardigheden, moet de robot tijdens het leren niet alleen de motorische aspecten beheersen, maar ook op de hoogte zijn van de context, zoals de locatie en vorm van het object. Daarom wordt in Hoofdstuk 7 de generalisatie benadrukt van de geleerde vaardigheid over verschillende contexten. De voorgestelde aanpak steunt op GP-theorie voor het verkrijgen van een niet-lineaire transformatie van de demonstratieruimte naar de uitvoeringsruimte terwijl onzekerheden behouden en gepropageerd worden. Experimenten met taken zoals oppakken en plaatsen van objecten, menselijke armen aankleden en oppervlakken schoonmaken demonstreren hoe de robot de uitvoering kan generaliseren door de attractor-, oriëntatie- en stijfheidsbeleid te transformeren naar tal van nieuwe scenario configuraties, zelfs zonder enige demonstratie van de vaardigheid.

In Hoofdstuk 8 wordt het concept van bewustzijn van onzekerheid uitgebreid naar het over-parametriseren van de context, zoals het volgen van meer objecten dan nodig. Het voorgestelde algoritme vraagt de aandacht van de gebruiker in het geval van dubbelzinnigheid, zoals wanneer meerdere gedetecteerde objecten het doel van de vaardigheid kunnen zijn. Feedback kan dan op verschillende manieren worden gegeven, bijvoorbeeld door de robot te duwen, te verplaatsen of door het geven van een beloning of straf. Uit een gebruikersstudie bleek ook de voorkeur van beginnende gebruikers om geen conventionele kinesthetische demonstraties te geven maar alleen in te grijpen wanneer nodig.

Sommario

Sebbene l'Intelligenza Artificiale (IA) sia sempre più utilizzata per automatizzare attività come la scrittura e la generazione di immagini, resta ancora difficile trovare risorse umane adeguate per compiti manuali come la gestione dei bagagli aerei o la raccolta di prodotti agricoli in serra. Tuttavia, la crescente necessità di adattare le capacità dei robot a contesti diversi, che spaziano dall'agricoltura alle faccende domestiche, richiede che i robot abbiano una morfologia versatile, come un braccio robotico, insieme a sensori avanzati e un'intelligenza sufficiente per adattarsi rapidamente a nuove situazioni. Nonostante la diffusione di video sensazionalistici online, la tecnologia robotica attuale non è ancora in grado di rispondere efficacemente a queste esigenze.

Il principale ostacolo che impedisce ai robot di eseguire compiti quotidiani, di fornire assistenza nei supermercati o di raccogliere frutta nei campi è la scarsità di dati necessari per costruire modelli affidabili del mondo reale. Di solito, lasciare che i robot esplorino autonomamente e determinino le migliori strategie non è considerato sicuro o pratico. Un approccio più efficace consiste nell'affidarsi alla supervisione umana. In questo caso, i robot possono chiedere chiarimenti quando sono incerti e gli esseri umani possono intervenire quando le azioni dei robot risultano errate o non all'altezza delle aspettative. Inoltre, quando il robot riceve istruzioni o ne richiede, dovrebbe essere in grado di quantificare il grado di confidenza nell'interpretazione delle correzioni. Questa tesi contribuisce al campo dell'apprendimento interattivo dei robot introducendo diversi algoritmi che tengono conto dell'incertezza, migliorando l'efficienza dei dati durante l'apprendimento e la sicurezza durante l'esecuzione.

Il Capitolo 2 introduce il lettore all'ambito dell' apprendimento interattivo e alle diverse modalità di interazione, dalle valutative a quelle correttive, sottolineando l'importanza della quantificazione dell'incertezza nella apprendimento di nuove abilità.

Il Capitolo 3 presenta le principali nozioni sui Processi Gaussiani, il principale metodo di apprendimento utilizzato in questa tesi per insegnare comportamenti ai robot e tenendo conto delle incertezze. Il Capitolo spiega come un Processo Gaussiano possa essere addestrato con i dati disponibili e le correzioni fornite dalle persone, e come si possano ottenere previsioni sia della media che della varianza delle azioni. Particolare attenzione viene dedicata all' aggiornamento dei modelli con correzioni ricevute in tempo reale e alla stima delle incertezze.

Il Capitolo 4 descrive come un robot possa apprendere compiti di manipolazione complessi tramite dimostrazioni interattive. In pratica, l'utente fornisce al robot una dimostrazione cinestetica, ossia lo guida fisicamente per trasferire la propria conoscenza su un'abilità specifica, come pulire un tavolo o inserire una spina in una presa. Gli esperimenti mostrano come la quantificazione e il rifiuto delle incertezze aiutino il robot a concentrarsi su aree in cui ha maggiore confidenza. Inoltre, l'aggiornamento continuo del modello consente di integrare le correzioni ricevute dall'utente, modificando così l'attrattore e la rigidezza,

garantendo che, ad esempio, la forza giusta venga applicata nella direzione corretta durante la pulizia di un tavolo.

Il Capitolo 5 estende l'apprendimento di un'abilità all'intera posa del robot e alle dita, esplorando come questo possa essere fatto con il minimo numero di dimostrazioni e correzioni. In particolare, al robot viene richiesto di eseguire la raccolta di oggetti in un movimento fluido senza fermarsi prima di chiudere la mano. Tuttavia, senza supervisione, l'esecuzione dell'abilità è spesso troppo lenta o l'oggetto cade prima che le dita si chiudano. Ciononostante, dopo aver ricevuto correzioni umane, i robot sono in grado di migliorare la rapidità di esecuzione e coordinare bene le dita della mana, riuscendo a raccogliere gli oggetti velocemente senza farli cadere.

Affidarsi esclusivamente alla posizione cartesiana corrente del robot durante l'apprendimento può risultare limitante, poiché non permette di codificare abilità che richiedono movimenti che si sovrappongono, ad esempio quando ci si avvicina un obiettivo e poi si torna lungo lo stesso percorso. Il Capitolo 6 propone quindi una nuova modalità di codifica dei comportamenti per insegnare abilità di manipolazione bimanuale, mantenendo un'interazione sicura con gli esseri umani attraverso la limitazione della velocità e della forza. Lo studio esamina anche l'efficacia delle correzioni cinestetiche, ovvero la possibilità di correggere abilità bimanuali semplicemente perturbando il robot in movimento. Insegnare o correggere due manipolatori contemporaneamente con dispositivi di teleoperazione può risultare complesso, perciò il metodo proposto esplora la possibilità di modulare i movimenti attraverso interazioni cinestetiche piuttosto che riapprendere completamente le abilità da zero ogni volta che il robot sbagli.

Nonostante l'efficacia dimostrata dai metodi proposti nell'apprendimento di abilità motorie singole e bimanuali, durante l'acquisizione di un compito il robot deve prestare attenzione non solo all'aspetto motorio, ma anche al contesto, come la posizione o la forma degli oggetti. Il Capitolo 7 affronta questo tema, enfatizzando la necessità di generalizzare le abilità motorie acquisite a contesti diversi. L'approccio proposto si basa sull' apprendimento di una mappa non lineare dallo spazio dimostrato a quello di esecuzione, mantenendo e propagando le incertezze. Gli esperimenti su compiti come la vestizione di persone e la pulizia di superfici mostrano come il robot sia in grado di adattare l'esecuzione modificando la posizione e l'orientamento a molteplici nuove situazioni a partire da una singola dimostrazione umana.

Infine, il Capitolo 8 propone un algoritmo che richiede l'intervento dell'utente in caso di ambiguità, come quando più oggetti rilevati potrebbero rappresentare la possibile cosa da raccogliere. L'ambiguità decisionale può essere risolta attraverso varie modalità di interazione, come spingere il robot, muoverlo o fornire ricompense e punizioni. Uno studio ha inoltre evidenziato che gli utenti inesperti preferiscono intervenire solo quando necessario piuttosto che fornire dimostrazioni cinestetiche tradizionali.

1

Introduction

Robots should represent sophisticated technological creations, acting as agents that perform tasks independently. Equipped with sensors, processing units, and actuators, they should follow instructions, respond to inputs, and work without constant human guidance. They should help us in tasks like surgery [64], cooking [160], handling the luggage in and out of airplanes [80], or harvesting fruit and vegetables in greenhouses [138]. This requirement is still not addressed with the current technologies, despite click-baiting videos shared online by multi-billionaires [113]. As a matter of fact, the progress made in robotics may not seem as substantial compared to the rapid advancements seen in language processing or computer vision. This is because robots still encounter challenges when attempting tasks that humans typically perceive as straightforward, such as manipulation [18].

The primary reason for the slower progress in developing embodied intelligence lies in the insufficient availability of data needed to train large models akin to those used in Large Language Models (LLM). Unlike language models, it is often impractical to instruct and correct a robot in a user-friendly manner, such as through touch or speech. Moreover, a significant issue arises from our low tolerance for their errors: misinterpreting instructions or experiencing hallucinations could lead robots to potentially dangerous interactions with humans and the surrounding environment. Safety concerns will significantly restrict the adoption of robotics in human-centric environments like daycares or supermarkets.

Furthermore, the robot typically lacks awareness of changes in task circumstances, such as alterations in object locations and shapes. This lack of awareness prevents effective generalization and recognition of situations, as the robot may not realize when it encounters unfamiliar conditions and may require guidance.

This thesis tackles the challenge of enhancing robots' intelligence by boosting their awareness of their knowledge boundaries - what they comprehend and what they lack - particularly when learning from interactions with human teachers.

1.1 TEACHING ROBOTS

When humans are available to share their expertise in guiding contact-rich manipulation tasks, it is preferable to encode behaviors using recorded demonstrations rather than opting for alternative machine learning approaches such as reinforcement learning (RL). The latter methods entail supplementary demands related to design, infrastructure, safety, and data efficiency [155] and often prove impractical for real-world physical systems due to constraints on time and resources. Conversely, Imitation Learning (IL) entails the robot acquiring knowledge directly through supervised methods from human trainers. This approach enhances learning efficiency and diminishes the likelihood of unsafe explorations.

1.1.1 IMITATION LEARNING

IL has obtained considerable attention as a potential direction for enabling all kinds of users to easily program the behavior of robots. In this context, humans can easily teach the desired skill to the robot using teleoperation or kinesthetic teaching, i.e., by dragging the robot around, and the skill can be transferred to the robot in a supervised learning formulation. According to the policy formulation, the robot learns what is the desired action to take with respect to the current input which could be the current time step, the current robot position, or the current image input.

However, when considering the advantages of programming robots in a natural way, as we humans do for teaching complex skills to each other, the possibilities are not limited to showing demonstrations and learning from them [11]. In particular, learning a skill from a demonstration does not guarantee that the result would resemble the desired intended behavior. For example, when teaching a robot how to clean a surface, how can the teacher know that the robot is accurately capturing the necessary need to apply a strong force when in contact with the surface? Moreover, foreseeing and demonstrating the task for every possible surface would be impractical even if the correct force behavior is captured from the demonstration.

Interactive Imitation Learning aims to allow users to provide additional instructions to the robot when necessary, particularly in situations where the robot's actions may be incorrect or pose a risk.

1.1.2 INTERACTIVE IMITATION LEARNING

Interactive Imitation Learning (IIL), conceptually summarized in Fig. 1.1, relies on different types of teaching modalities, like demonstrations, sporadic corrections, or evaluations (grading) with value judgments or rankings. For humans, this kind of interactive teaching approach seems to be the most natural strategy for teaching to perform more complex skills.

For example, if, after the demonstration, the robot is not applying sufficiently strong force, the user could give directional teleoperated feedback to the robot and explicitly tell it to push harder. Moreover, teaching complex skills, such as picking objects at non-zero-velocity that would be prohibitive to be demonstrated by novice users, becomes significantly easier if the original slow demonstration is interactively corrected using human feedback on the desired end-effector velocity. This enables the user to improve the robot's behavior over observed mistakes. Human teachers can transfer their knowledge to the learning agent through different modalities of interaction, see. Ch. 2, and they are able to observe the effect of their feedback throughout the incremental learning of the skill. Moreover, the robot can

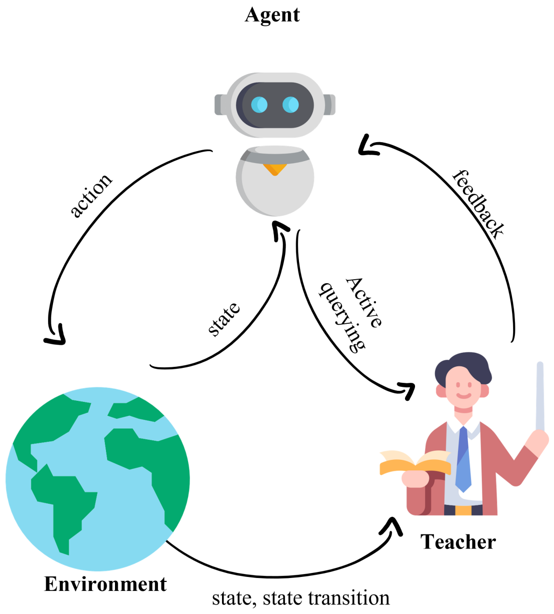


Figure 1.1: A general schema of Interactive Imitation Learning from [38]

also detect uncertainties in the tasks and actively query the user help [150] aiming to increase learning efficiency. This thesis focuses on the uncertainty quantification of low-level and high-level policies when learning from interactive demonstration, making the robot aware of what *it knows* and what *it does not* and how to effectively and efficiently integrate human feedback.

1.1.3 Uncertainty-aware IIL

The aim of Uncertainty-Aware IIL is to have robots that, thanks to the uncertainty quantification, become safer around humans, require less data to be trained, update more efficiently their beliefs from non-expert human feedback, and be robust to eventual mistakes in the correction. The main requirement is that, given a certain situation, the robot must predict the mean and variance of the action given the evidence of what was observed during the demonstrations. By leveraging the awareness of uncertainty, the robot can enhance its performance in learning and refining motor skills quickly. It achieves this by actively seeking the teacher's attention through queries or by looking for areas of workspace with lower uncertainty, and efficiently updating the policy upon receiving corrections.

1.2 Contributions

Interactive Imitation Learning combined with uncertainty awareness promises to accelerate the development of robot learning and adaptation in complex manipulation tasks. However,

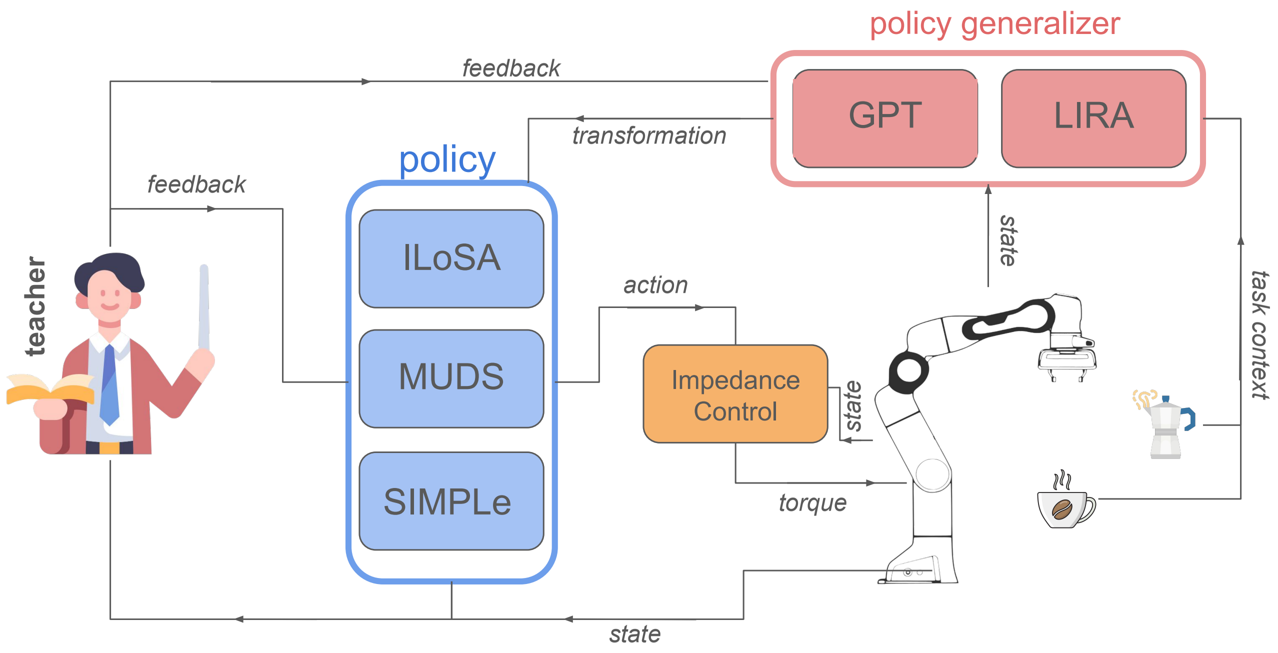


Figure 1.2: General scheme of this thesis.

there is not enough study in the field due to technical and scientific challenges. Hence, we can formulate the following research question:

Research Question : 1

What practical implications and performance improvements arise from integrating uncertainty quantification/exploitation, tailored for interactive learning scenarios, in tackling the challenge of instructing robot manipulation tasks? How does the performance of these methods compare to that of existing approaches that do not incorporate any feedback modality or consider uncertainties?

To answer this question, different machine learning methods were proposed to tackle challenging manipulation problems, like assembly, disassembly, and picking at non-zero velocity, with a single or double arm setup. Moreover, the methods were tested with non-expert users, and the usability of the methods was evaluated to also answer the following question:

Research Question : 2

How does the choice of teaching strategy (full demonstrations vs. corrections) and the incorporation of uncertainty/ambiguity awareness affect the non-expert user experience?

The next section introduces the developed methods, the application, and the new usability insights that were obtained during the development of this thesis.

1.2.1 Methods

Fig. 1.2 shows the interactive learning scheme that is used in this thesis. We notice the three important elements that are also present in Fig.1.1, i.e. the agent, the teacher, and the environment. However, the agent is split into policy and policy generalizer. From a biological perspective, the policies represent the motor cortex that is involved in the planning, control, and execution of voluntary movements; on the other hand, the policy generalizer is the parietal lobe that is involved in processing sensory information and integrating it with higher-level cognitive functions such as spatial reasoning and attention, which are essential for conceptual understanding. Each of the submodules represents an alternative solution for the motor policy and the task generalizer.

ILoSA, **I**nteractive **L**earning **o**f **S**tiffness and **A**ttractors, introduced in Chapter 4, allows the user to interactively correct the attractor field and a Cartesian impedance control's stiffness matrix to match the necessary high/low forces locally. The recorded offline demonstration would not have enough information to successfully apply high/lower forces at the right time when dealing with contact-rich manipulation like plugging, unplugging, cleaning, or pushing. The algorithm proposes a straightforward way of disambiguating the directional corrections on the attractor, stiffness, or both.

MUDS, **M**inimum **U**ncertainty **D**ynamical **S**ystems, introduced in Chapter 5, encodes the provided demonstration as a dynamical system. To avoid the issue of covariate shift, when the robot is dragged out of the distribution of the demonstrated data, a vector field proportional to the gradient of the estimated epistemic uncertainty is superimposed to the fitted dynamics, resulting in the local stabilization motion close to the region of low uncertainties. During execution, if directional human feedback is received, the algorithm efficiently updates the learned dynamics matching the human intended behavior. This enables reshaping the motion when the objective of the task is changed or incrementally updating the policy with the human in the loop.

SIMPLe, **S**afe, **I**nteractive **M**ovement **P**rimitives **Le**arning, introduce in Chapter 6, enables the establishment of safety limits on the resultant velocity and force field of the manipulator's Cartesian motion. A novel encoding of the skill also facilitates the teaching of manipulation tasks with long time horizons and provides the capability to iteratively adjust the motion through kinesthetic corrections. The algorithms enable instructing and modifying bimanual manipulation policies by teaching one arm at a time or both arms simultaneously and interactively fine-tuning the manipulation strategy.

GPT, **G**aussian **P**rocess **T**ransportation for Policy Generalization, introduced in Chapter 7, proposes a method to transform policy from one task parameterization space shown in the demonstration to the space that is faced during execution. The task parameterization could be the object and goal position in a pick-and-place scenario or the point cloud of the surface to clean. This method allows the end-user to teach a policy in one task and automatically generalize it to different arrangements of the scenario.

LIRA, **L**earning **I**nteractively to **R**esolve **A**mbiguity, introduced in Chapter 8, proposes an active learning framework to interactively query the user when facing an ambiguous situation, i.e., when more than one action would be equivalently valid, given the evidence of the previous demonstration or corrections. It is intended to learn multi-sequence, multi-frame pick and place tasks, where for every segment, the robot has to disambiguate which is the correct object to pick and where to correctly place it. Multiple feedback modalities are possible, simple yes/no, directional feedback, or local kinesthetic teaching.

1.2.2 NEW HUMAN USABILITY INSIGHTS

Beyond the formulation of novel algorithms and methods that allowed the transfer of knowledge to the robot, this thesis also investigated the usability of the methods from non-expert users. Some studies focused on the validation of the method itself or also comparing different teaching modalities, such as interactive versus not or different feedback modalities. However, first and foremost, we want to answer the question: *"Can anybody, with no robotics or machine learning background, be able to correctly teach complex manipulation skills to the robot?"* In the thesis, the reader will find out that:

- users can teach how to insert a plug in a socket with kinesthetic demonstration and teleoperated corrections thanks to the efficient attractor and stiffness update proposed in **ILoSA**. The study is part of Chapter 4.
- users prefer to demonstrate at a lower speed and use interactive corrections when asking to teach to perform a picking task as fast as possible, with the goal of matching a requested performance. The study is part of Chapter 5.
- user prefers to locally reshape a manipulation task with kinesthetic teaching rather than provide a completely new demonstration when asked to teach to pick a different object with a bimanual setup. The study is part of Chapter 6.

- users prefer to teach robots by giving interactive corrections rather than aggregating more and more demonstrations in the context of reference frame disambiguation. The study is part of Chapter 8.

1.3 Applications

All the applications are lab reproductions of real-life scenarios, showing a tangible impact on industrial and daily life applications, e.g., using real plugs and sockets. Having realistic objects allows us to make a step closer to real-world impact and one step away from conceptual peg-in-the-hole tasks.

1.3.1 Household chores

The integration of robotics in automating household chores is poised to become a vital solution for addressing the needs of an aging population. As our society ages, the demand for caregiving services is on the rise, and robotics offers a promising avenue to alleviate some of the burdens associated with elderly care. These advanced robots can assist with tasks like cleaning, cooking, and even personal care, providing independence and dignity to seniors. However, many of these tasks require the learning of a contact-rich manipulation. In this thesis, examples of a robot learning how to insert/extract plugs, see Fig. 1.3, cleaning tables, see Fig. 1.4 and various different objects, see Fig. 1.5, are studied in the context of IIL.

1.3.2 Retail automation

In recent years, grocery stores have undergone significant transformations. Self-checkout options have become increasingly prevalent, allowing customers to scan and pay for their purchases without waiting for an available cashier. However, all the tasks that guarantee to find goods available on the shelves are still not automated. In particular, in the last stage, after products arrive at the supermarket, they need to be correctly allocated in the right place. In this thesis, many challenges are highlighted and tackled in the experiments, such as learning how to re-shelf as fast as possible using one fast movement while not flipping the object down, see. Fig. 1.6, or how to effectively sort the fruits in the right box, see Fig. 1.7,

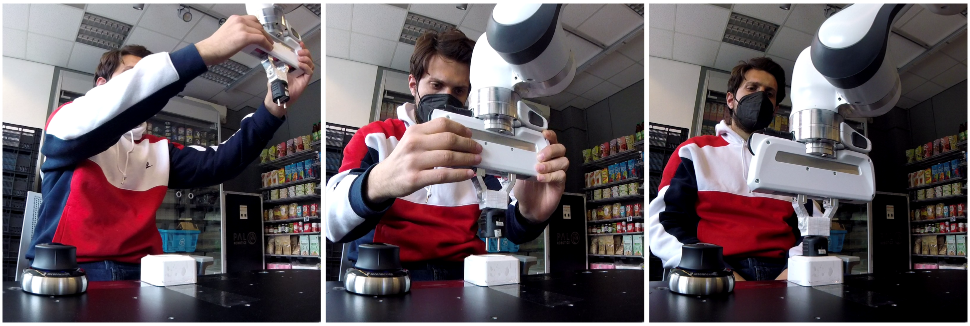


Figure 1.3: Human teaching how to successfully insert a plug into an outlet. This can be useful in the context of teaching robots how to use home appliances, e.g. vacuum cleaners. Experiments related to Ch. 4.

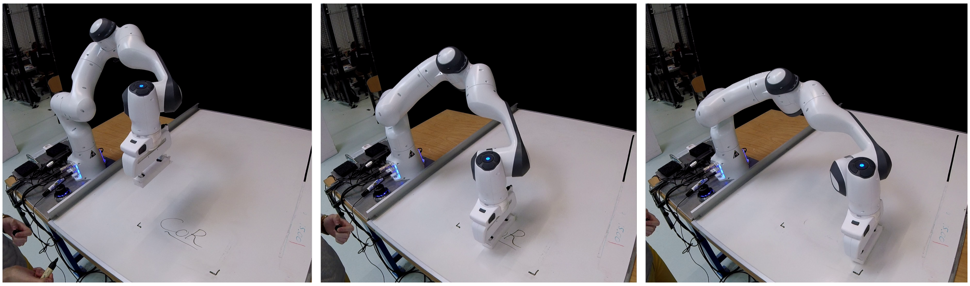


Figure 1.4: Robot cleaning a dirty table after receiving kinesthetic and corrective feedback from the human demonstrator. Experiments related to Ch. 4.

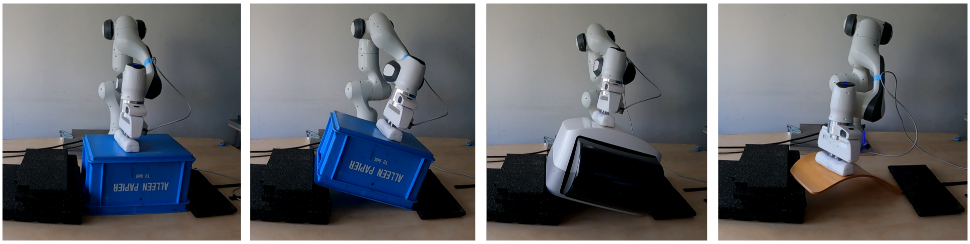


Figure 1.5: The robot generalizes the cleaning strategy to different objects after perceiving them with a depth camera. Experiments related to Ch. 7.

or handling heavy objects like crates, relieving workers from this wearing task, see Fig. 1.8.

1.3.3 Industrial Assembling/Disassembling

Increasing retail and consumption also lead to global challenges, such as mounting electronic waste by the rapidly accumulating discarded electronic devices and the associated environmental and health hazards. Robotics emerges as a pivotal solution to combat this problem. E-waste poses a critical issue due to its sheer volume and complexity, containing many materials and components. This complexity makes manual disassembling a labor-intensive and time-consuming endeavor. Robotics, however, can significantly expedite and streamline

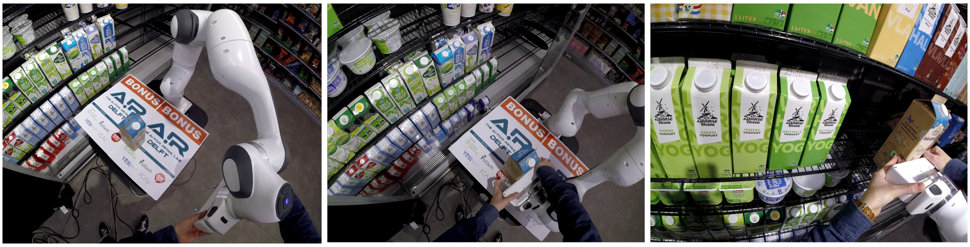


Figure 1.6: Point of view of a teacher showing a robot how to pick and place a milk box in one single movement in the context of reshelving items in a supermarket. Experiments related to Ch. 5.

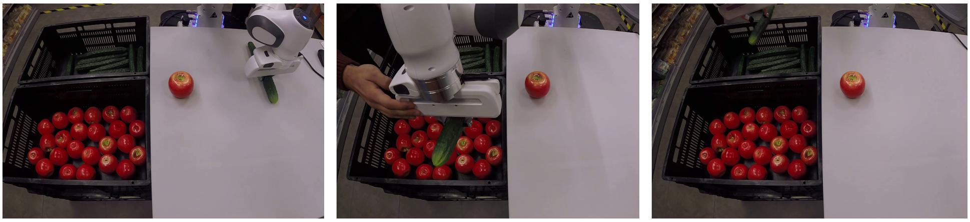


Figure 1.7: Robot learns from a human teacher how to correctly allocate the vegetable in the right crate in the mundane task of fruit sorting. Experiments related to Ch. 8.

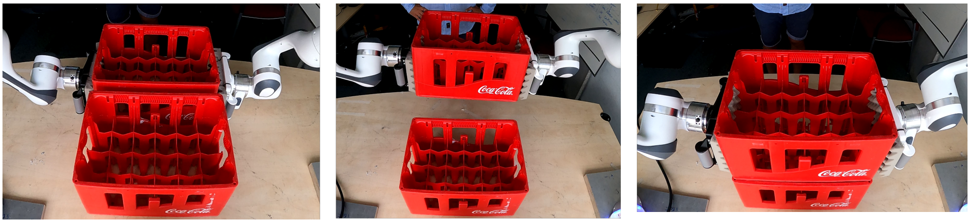


Figure 1.8: Bi-manual robots taking over the physically demanding role of lifting heavy items in a warehouse, relieving humans from the strain on their backs. Experiments related to Ch. 6.

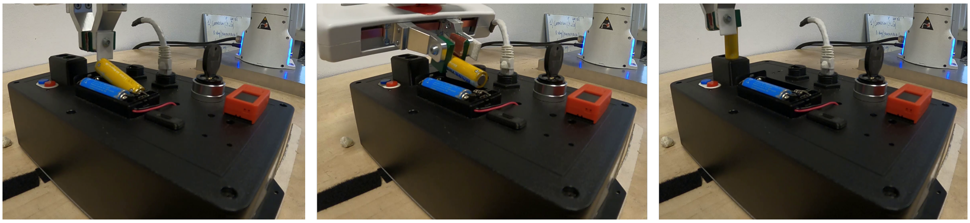


Figure 1.9: Example of precision battery removal from the electronic device, for the sake of disposal. Experiments related to [118].

the recycling process. Figure 1.9 shows the execution of the robot that learned from a human how to accurately extract a battery, grasp it, and then place it in the right container, relieving the human from the task.

1.4 Thesis Outline

The thesis is organized as follows:

- Chapter 2 presents the state of the art on Interactive Imitation Learning and the different teaching strategies that can be used to teach robots, using human reward or preferences or absolute/relative corrections in the state/action space.
- Chapter 3 summarizes all the mathematical background needed to understand how Gaussian Process Regression is used and implemented in the thesis. Novice readers of the topic should understand at least the basics of the topic and be introduced to the more advanced implementations of the algorithms of the thesis.
- Chapter 4 presents the Interactive Learning of Stiffness and Attractors, a machine learning framework that allows the robot to learn complex manipulation tasks from interactive demonstrations.
- Chapter 5 extends on Chapter 4 and studies how learned dynamical systems policies of full robot pose and gripper can be interactively shaped with corrections to perform picking task in a single fluent motion.
- Chapter 6 presents the Safe Interactive Movement Primitives Learning that allows users to teach single or bimanual manipulation tasks with long time horizons while being safe around humans with constrained velocity and force actions.
- Chapter 7 formalized the concept of Gaussian Process Transportation to generalize policies to different task parametrization spaces while retaining the uncertainty quantification from the original policy and due to the transportation process. It was tested in generalizing various manipulation tasks like pick-and-place, cleaning, and dressing.
- Chapter 8 presents the Learning Interactively to Resolve Ambiguity, used to actively query the user attention when facing an ambiguous situation, i.e., when more than one detected frame can be used to generalize the original policy.
- Chapter 9 concludes the thesis and discusses future work.

2

Teaching Modalities in Interactive Imitation Learning

Along the development of this thesis, whether we are learning the action policy or the generalization policy, the primary source of information is the (human) teacher, see Fig. 1.2. We assume that the teacher is not only providing demonstrations, for instance, by moving the robot around or teleoperating the robot, but they also can provide feedback on the learned performance of the robot, such as correcting it and improving its performance. For example, the user can specify that an object is not the proper object to pick, that the current pressure applied to a surface has to increase, or that it has to perform a task faster.

This chapter categorizes different state-of-the-art methods by exploring how teachers can interactively train agents by providing feedback to them. Here, feedback refers to information explicitly conveyed by human teachers to learning agents via a Human-Robot interface.

We can classify these methods into two primary groups based on the teacher's feedback type: evaluative space and state-action space. The former assesses agent performance, i.e., tells a robot if something is good or bad, while the latter guides task execution, i.e., explicitly tells the robot what to do. In both categories, teachers can provide assessment or guidance through relative and absolute methods. Relative feedback indicates the direction for the agent's behavior change relative to the current or other policy executions, while absolute feedback conveys the teacher's knowledge of optimal behavior. Relative feedback is less informative but demands less cognitive effort from teachers, possibly sacrificing data efficiency. Hence, the choice between relative and absolute feedback involves a trade-off between data efficiency and teacher cognitive load during interaction.

This chapter is partially based on 🗎 *Celemin, C., Pérez-Dattari, R., Chisari, E.,* ***Franzese, G.****, de Souza Rosa, L., Prakash, R., Ajanović, Z., Ferraz, M., Valada, A. and Kober, J., 2022. Interactive imitation learning in robotics: A survey. Foundations and Trends® in Robotics, 10(1-2), pp.1-197 [38].*

This thesis primarily addresses Interactive Imitation Learning, where robots learn not only from offline demonstrations but also from online correction and evaluation. This background chapter will overview the current state-of-the-art in this area, including its limitations when applied to real human feedback or when quantifying uncertainties in the learned policy during deployment. The chapter is divided into three sections:

1. Sec. 2.1 highlights the meaning of learning from human reward and human preference, i.e., no specific action label needs to be provided to the robot but only rewards or by specifying which option they would prefer among two (or more). This modality is the least data and time-efficient way of teaching a skill, so it was discarded from comparisons and development of the thesis. A reader in a hurry could jump over this section without losing context.

2. Sec. 2.2 introduces interactive learning within the state-action space. During the interaction, users may label desired actions, such as specifying robot velocity or force at a particular position. However, providing exact values may be challenging for general users. Therefore, the section also discusses providing relative feedback, like indicating if it should be faster/slower or push harder/softer. Additionally, it reviews the distinction between robot-initiated and human-initiated feedback: robots may seek feedback in uncertain situations, while users may request policy adaptations to meet performance requirements.

3. Sec. 2.3 will conclude and discuss what highlighted in this chapter.

2.1 HUMAN FEEDBACK IN EVALUATIVE SPACE

The earliest interactive learning endeavors fall under evaluative feedback and draw inspiration from animal clicker training, a method commonly employed to train dogs and other pets [131]. Training animals for tasks like assistance or detective dogs demonstrates humans' ability to convey knowledge to other agents through straightforward signals indicating the acceptability of behavior, without explicit demonstrations of the task's execution, as seen in traditional methods of Learning from Demonstration (LfD). The feedback can be absolute or relative evaluations of performance. For example, where the teacher explains the right way of solving an exercise (absolute) or describes how some executions are better or worse than others, using either pair-wise comparisons or rankings (relative).

2.1.1 LEARNING FROM HUMAN REINFORCEMENTS

Utilizing evaluative feedback from a human teacher simplifies two challenges in contrast to autonomous learning methods such as Reinforcement Learning (RL). It eliminates the need to tackle the complex task of designing an objective function for autonomous feedback and streamlines system implementation by removing the necessity for reward computation infrastructure.

Thomaz and Breazeal [156] showed how Interactive Reinforcement Learning (Interactive RL) enables a human user to provide positive and negative rewards in real-time in response to robot actions and to advise anticipatory guidance input that constrains action selection choice and guides the learner towards the desired behavior. Since a human reward may have a different meaning with respect to an encoded environment reward function, which

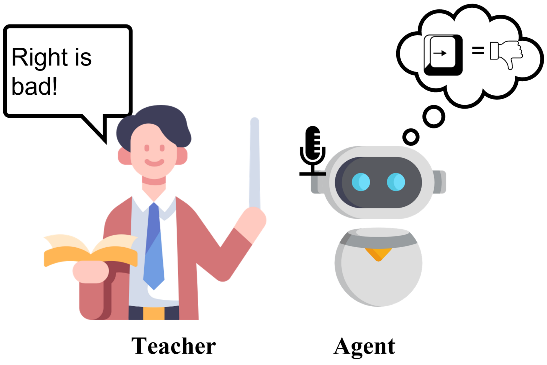


Figure 2.1: Learning from human reinforcements loop: the teacher is teaching the robot to go to the left and he gives bad rewards when it goes right.

is the basic reinforcement used in the conventional RL approaches, a series of works have analyzed how to model the human reinforcement [157, 158]. For example, the Training an Agent Manually via Evaluative Reinforcement (TAMER) framework [90, 91] addresses how to use delayed human rewards in RL problems with discrete action spaces. However, rather than using the evaluative human feedback as a reward, it can also be used to directly update the policy [66]. The policy is trained by increasing or decreasing the probability of an action in a certain state, depending on the feedback provided. Additionally, Loftin et al. [107, 108] propose a method to take into account the user feedback strategy, in particular taking into consideration different interpretations of lack of feedback from the teacher.

Human reinforcement enables teachers to convey insights about what is right or wrong at each time-step to the agent. It demands a solid grasp of the task without requiring expertise or knowledge of precise actions in every state. However, since this feedback does not explicitly specify alternative actions when a punished action occurs, a single erroneous punishment necessitates numerous new feedback instances to correct the impact of the incorrect feedback. This makes these approaches less robust when dealing with imperfect teachers.

2.1.2 Preference-Based Policy Learning

When learning a behavior, the human teacher is conveying the desired way of performing a task. For example, when picking an object, the robot can decide to approach it from left or right, see Fig. 2.2. Hence, without having any prior information, the robot can execute both trajectories and ask the user if they prefer one way or the other.

Methods for learning from human preference consist of comparing two or more sequences of actions and providing a preference score to the agent, and they do not require the teacher to identify and evaluate what is the credit of the decision at each time step with respect to the success or failure of the task execution, i.e., potentially reducing teacher workload. In other words, they use relative evaluative feedback that implicitly indicates the direction in which the solution in the policy space should be shifted, such that it matches the preferences

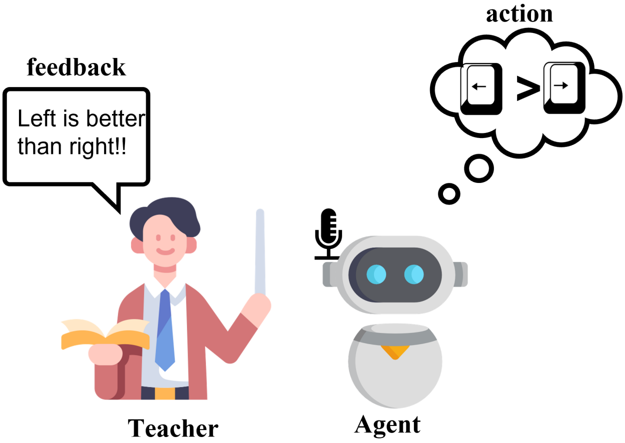


Figure 2.2: Learning from human preferences: the teacher is teaching the robot to take the left turn and it is specifically saying that going left is better than going right.

of the teacher. However, since this feedback is relative to other trajectories, policies, or roll-outs, it does not describe how good execution is in general, and a policy that is preferred over another or ranked as the best out of a set of policies might be ranked low later on with respect to some different executions.

Preference-Based Policy Learning (PPL) [3] is one of the first methods to integrate preference learning and RL. A human teacher provides feedback as pairwise preferences between policies, and the agent estimates the value of parametrized policies, selecting another set of policies in an iterative process. Fürnkranz et al. [58] also explore preference-based reinforcement learning, focusing on action preferences in a given state. Akrour et al. [5] extends this work by accounting for human mistakes. Jain et al. [78, 79] propose a co-active online learning framework where the human teacher provides small adjustments to system-generated trajectories. The reward function used in the RL step is learned via preference feedback, outperforming demonstrations or preferences used in isolation. The most important and critical component of this method is the choice of trajectories to compare. Generating safe and informative trajectories is a non-trivial limitation of these approaches. Active preference-based methods aim to improve convergence by generating informative queries [4, 124, 173], and the preference can also be used to learn a reward function [19, 43, 142].

Learning from preferences reduces the effort of the teacher and widens the spectrum of people who could teach a robot. Teachers do not need to be experts on a task to point out the best among two to multiple solutions. However, preferences are a relative measure of performance that evaluates a sequence of transitions, therefore the feedback does not specify what decisions make one roll-out better than the other, and the algorithm has to identify them while compromising data efficiency. Moreover, methods based on learning from preferences are also sensitive to mistakes in the teachers' assessments. The mistakes in the feedback have a negative impact on the convergence of the process, reaching lower policy performances.

A better, more data-efficient, and effective solution is to directly give feedback on the best action to take, highlighted in the following section.

2.2 HUMAN FEEDBACK IN STATE-ACTION SPACE

Human feedback on state-action space directly guides the robot on *how to perform* the task, offering explicit instructions on the right action or state transition. Unlike learning from evaluative feedback, this type of feedback does not involve an explicit quality assessment of the policy; instead, it conveys the teacher's insights and understanding of task execution. This feedback can be absolute, where the teacher demonstrates the optimal action, e.g., moving a robot on a new trajectory, or it can be relative, where the teacher only specifies in which direction the action should be changed, e.g., by telling to go faster or slower at a certain point in space. However, it does not assume that the correction represents the optimal action but rather serves as a hint in that direction. The correct action is eventually reached through an iterative accumulation of incremental progress from multiple relative corrections.

2.2.1 LEARNING FROM ABSOLUTE CORRECTIONS

In this interaction, agents are supposed to receive explicit task execution demonstrations from the teacher, with the learning policy concurrently guiding the agent. Depending on the method used, the teacher can offer corrective demonstrations at each time step, sporadically based on their judgment, or in response to queries from the learner (active learning). These methods are the closest to standard LfD methods like Behavioral Cloning.

Some of the most important methods for learning from corrective demonstrations are inspired or belong to a family of approaches based on Data Aggregation (DAgger) [139], which interactively records the correct action demonstrations while a novice policy is controlling the agent. The idea behind DAgger is to iteratively generate roll-out trajectories with the current policy, query the expert for corrections on the visited states, and finally aggregate the recorded teacher actions to the dataset. As with other methods in this section, this approach enables the expert to provide corrections on the states visited by the current policy. This may lead the robot to new places in the workspace and the new aggregated teacher corrections can be used to learn how to recover from them. However, if the expert is a human, this continued aggregation is often unfeasible and prone to incorrect labels for robotic tasks, which usually operate at high control frequency and the exploration of unknown regions may result in unsafe and undesired behaviors.

A possible alternative is to monitor the policy execution and intervene when necessary, taking over control of the robot completely. This setting can be defined as learning from human intervention, and numerous studies have been presented to investigate such methods. There exist two main types of human intervention approaches: Human-Gated, i.e. the human decides when to intervene and Robot-Gated when the robot decides [101].

HUMAN-GATED INTERVENTIONS

Human-gated interventions allow the expert users to decide when to intervene (control the robot). For instance, if a robot is attempting to insert a plug in the socket but is diverging away from its target, the human teacher can take control of the manipulator and show, by teleoperation or kinesthetic teaching how to perform the task. This ensures safer policies given that the expert is always ready to intervene in case of dangerous behaviors.

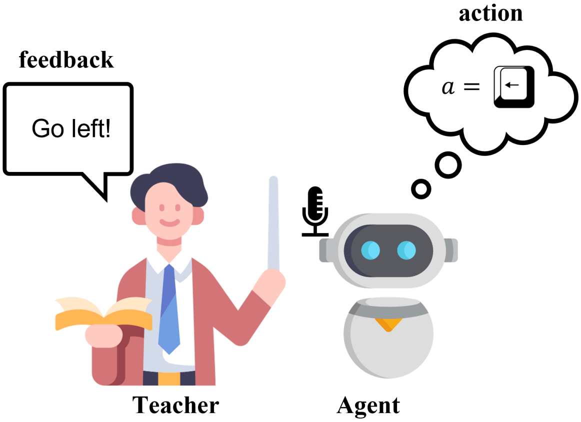


Figure 2.3: Learning from human absolute corrections: the teacher is explicitly telling the robot to go left.

Human Gated DAgger (HG-DAgger) [87] is a direct extension of the DAgger algorithm, where the human teacher is in charge of intervening when the agent drifts away from the desired behavior. Every time an intervention occurs, the expert trajectory is recorded and stored in the training dataset used to optimize the policy. Additionally, HG-DAgger learns a safety threshold of a risk metric, which could be used as a policy confidence metric for different regions of the state space.

Another recent work in the same category is Super-Human InsErtion using Learning from Demonstration (SHIELD) [109], which focuses on the problem of industrial insertion. It extends the Deep Deterministic Policy Gradient from Demonstration (DDPGfD) [168] algorithm with on-policy corrections, i.e., the human can intervene to guide the agent back into the optimal region in case of deviations.

Finally, interventions can also be combined with demonstrations. For example, in Cycle-of-Learning [63], human-gated interventions are used for improving a policy obtained from demonstrations recorded in a warm-up stage. The experiments showed that this approach has better performance than using either only demonstrations or only human interventions.

However, the main challenge when aggregating new labels online is the (very) delayed effect on the policy learned policy leading to possible dangerous behaviors, e.g. the robot does not understand on time that it has to steer and collide with a human or a wall. This is one of the main challenges that are addressed in this thesis where in Ch.3 we formulated a new model update and aggregation rule that has an immediate effect on the learned policy.

Robot-Gated Interventions

Robot-gated interventions require the agent to estimate when an intervention is necessary, which does not require constant attention from the teacher, since the robot is the one deciding when the intervention should be performed, allowing the human to supervise multiple robots at once [74]. These methods generally require the agent to estimate a measure of performance, safety, or uncertainty about the currently observed state, which is then used

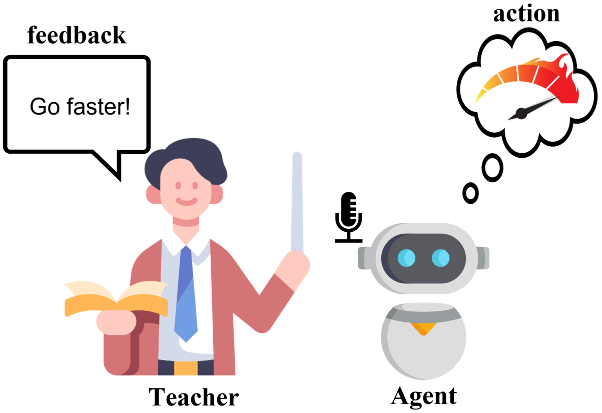


Figure 2.4: Learning from human relative corrections: the teacher is correcting the velocity of the robot, telling it that it can increase the value with respect to the current one.

to determine when to query or enable human teacher control. However, these kinds of approaches have to deal with the disengagement of the users, who do not react immediately when requested and require some time to be able to take over the system again.

A variation of DAgger called Safe DAgger (SafeDAgger) [177] trains a classifier that predicts whether the learning policy deviates from the expert and if it is the case, it switches the control to the expert in order to prevent executing unsafe actions. On a similar note, Ensemble Dagger (EnsembleDAgger) [114] relies on a doubt rule that also switches control from the learning policy to the expert teacher. The doubt rule is computed based on the novelty/uncertainty of the policy, which is measured with the output variance estimated with an ensemble of neural networks.

Finally, Thrifty DAgger (ThriftyDAgger) [74] proposes to query interventions in case the encountered state is sufficiently novel or risky. Similar to EnsembleDAgger, novelty is estimated as the output uncertainty, whereas the "risk" of a state is estimated by learning an exacted cumulative reward function to evaluate the discounted probability of success from that given state and the action proposed by the policy.

The estimation of action uncertainties can significantly improve the robot's awareness and query the teacher to provide more demonstrations or corrections. However, in the previously mentioned works, the uncertainty is usually estimated with ensemble methods that may be uncalibrated, e.g. query the user too much or too little, hence we will formulate all the interactive policies directly relying on a fully Bayesian method, i.e. Gaussian Process, as described in Ch. 3.

2.2.2 Learning from Relative Corrections

Imagine that you are teaching a robot to execute a task like picking an object and you want to increase the execution speed in a certain location of the space. Asking the teacher to provide the exact velocity that the robot should execute is impractical since no one, not even an expert teacher, will be able to provide a precise number that would effectively execute the picking task. An alternative is to incrementally increase the velocity by giving corrective actions and observing the execution performance of the robot, see Fig. 2.4.

In essence, the teacher should be capable of providing a rough estimate of how a transition would change if the policy underwent slight modifications. For example, knowing that reducing power in a propeller decreases acceleration or increasing force on the brake pedal slows down a car. This correction could be quantized as well as continuous-valued, depending on the interface used. For example, when using buttons we can set a default increase for each button, or then using a joypad, we can have a continuous value that can be given as a correction.

COrrective Advice Communicated by Humans (COACH) [37] framework employs binary feedback to indicate, for a given state, the direction in which the action taken by an agent has to change, while the magnitude of the change is set as a predefined parameter in the range of the actions. The feedback provided in COACH and the policy updates occur while the agent is interacting with the environment, i.e., during policy execution time, which allows the teacher to directly observe the effects of the corrections and correct again if required, speeding up the learning process.

Some works that are more focused on teaching behaviors with manipulators have been proposed for letting the teachers provide kinesthetic corrections over the executed trajectories. These relative corrections are used for either updating a policy or updating the objective function that can be used in a model-based setting with a planner system. For instance, a policy correction by the teacher on the end-effector displacement with respect to the original trajectory is detected with tactile sensors in Tactile Policy Correction (TPC) [12]. The correction could be used for policy refinement or policy reuse. In the former, the corrections are added as new data points to the training set, whereas in the latter the corrections are used to replace some already existing data points.

Additionally, incremental refinement of trajectories of context-dependent policies is performed with kinesthetic feedback in [48]. The corrections are not detected and computed with tactile sensors, but rather with the measured position difference between the desired trajectory and the one disturbed by the teacher. In Canal et al. [36] kinesthetic corrections are also used to reshape a movement primitive used for a feeding assistance robot application.

2.3 DISCUSSION

Interactive methods enable teachers to train agents that outperform policies generated through standard IL. They are particularly effective at mitigating the issue of lack of informative data because they incrementally collect more comprehensive data through teacher interventions during or after policy roll-outs.

Some studies have compared different interaction modalities and found that users generally prefer methods where they communicate information explaining or demonstrating how to perform a task rather than simply assessing policy quality [10, 95, 158, 161]. However, the choice of modality depends on the task, the teacher's expertise, and the available interfaces.

The choice of modality affects the *inclusivity* of potential teachers based on their expertise. Corrective demonstrations provide the richest feedback, requiring highly skilled teachers. Relative corrections widen the pool of potential teachers since they do not need to be experts and can suggest incremental improvements. Given the focus of this thesis to contribute to interactive learning methods and to prove the higher inclusivity with respect to non-interactive methods, a series of studies with non-expert users were performed.

Moreover, various factors, such as physical constraints (e.g., real physical robots), time limitations, and available interfaces, must be considered when selecting the appropriate modality. For example, the choice of teaching interfaced, e.g. by physical perturbation or by teleoperated feedback, affects the user experience, no matter the encoding of the policy.

Nevertheless, the robot's awareness of uncertainties in the data and the model is a key point to reduce the number of required corrections that the user has to provide. By reformulating the learning policies with a clear statistical formulation, the user may need less feedback to communicate desired changes to a robot, and in case of user unavailability, the robot could also correct its own behavior by preferring the actions that will decrease its uncertainties.

The next chapter will introduce the foundations and the contributions on the machine learning side that are at the foundations of the greatest part of the algorithms proposed in this thesis.

3

Interactive Learning with Gaussian Processes

When using Interactive Imitation Learning, the policy is estimated from (finite amount of) data but is often required to act on a continuous space, e.g., when controlling a robot motor torque or to generalize in previously unknown states. Therefore, fitting a continuous function to approximate the data is necessary. Since we do not have an infinite amount of data, assumptions on the function need to be made, e.g., linear, non-linear, parametric, non-parametric, etc. Moreover, the nature of the regressor that it is used will mainly change the behavior of the agent when going out of distribution.

For example, let us imagine that we are teaching a robot how to clean a table by giving a set of kinesthetic demonstrations, i.e., by moving the robot around and recording the motion. The robot will eventually learn how to move as a function of its Cartesian position. However, we may encounter two hypothetical scenarios: the first is that the robot did not learn correctly the right action to take, e.g., it is not applying enough pressure on the table, and the second is that it is dragged into regions of space where it does not know what to do, e.g. far away from the table. Considering that attempting the wrong action can be (very) dangerous, in particular when surrounded by humans, we require the agent to estimate the uncertainty of its actions, rely on some prior knowledge, and quickly learn from human feedback.

A successful candidate to address all these requirements is a Gaussian Process (GP), which also learns the distribution of the robot policy in addition to the mean. The key advantage of having also information about the distribution is the increase in the trustworthiness of robot action and improved data efficiency in training and during online updates. In this chapter and throughout the rest of the thesis, different formulations of GPs are proposed to address the goal of obtaining an uncertainty-aware interactive imitation learning process.

In this chapter, the reader will be first introduced to Gaussian Processes and how they are used in the context of interactive robot learning of motion skills or skill generalization. In

- Sec. 3.1, the idea of fitting a GP by finding the posterior distribution given a prior and the evidence of the collected data is introduced. The prediction function for the mean and the variance are used during the thesis to make predictions with the models, for example, to predict attractors or stiffness of a robot manipulator, in Ch. 4, or the deformation map for a generalization policy, in Ch. 7.
- Sec. 3.2, two rules are proposed to update the models online when new corrective data are provided to match the current circumstances, such as when learning to go faster to pick an object at non-zero velocity in Ch. 5.
- Sec. 3.3, the mean and the variance of the derivative function, fitted with a GP and the analytical calculation of the uncertainty rate (as a function of the input) are recalled. The properties are used in the thesis to, for example, pull the robot close to regions of minimum uncertainties, see Ch. 4 or to compute the derivative of a generalization map in Ch. 7.
- Sec. 3.4, variational approximation of the posterior to reject the criticism on the impossibility of scaling GPs to big datasets. The proposed methods and update rule would apply directly also if the posterior is the approximate one rather than the exact one that considers all the recorded data.
- Sec. 3.5, a multi-output GP model is formalized, explaining the possibility of sharing (or not) the kernel hyperparameters or the likelihood noise among the different outputs. Learning multi-inputs-multi-outputs models is going to recurrently happen in different chapters of the thesis when learning movement skills or generalization maps.

This chapter is not intended to be a self-contained explanation of Gaussian Processes, given the breadth of the topic and ongoing research in the field. Excellent resources, such as Gaussian Processes for Machine Learning [172], can complement the reading of this chapter. However, sections or chapters marked with the symbol † indicate that the content represents an original contribution of this thesis.

3.1 Exact Gaussian Process

A GP is a generalization of a Gaussian distribution over functions. In other words, a GP defines a distribution over functions, where any finite set of points from the function's domain follows a multivariate Gaussian distribution. If we want to find this distribution in a fully Bayesian way, then we must define a prior distribution over all the possible functions.

3.1.1 Prior

The prior distribution represents our beliefs about the functions before observing any data. The prior is typically specified as a mean function and a covariance function.

Our prior belief is that our function is a sample of a GP defined by a mean function and a kernel matrix, i.e.,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')),$$

where $k(\boldsymbol{x},\boldsymbol{x}')$ is the kernel function that defines how much the function value at a certain input, $f(\boldsymbol{x})$, correlates with the function value at another input, $f(\boldsymbol{x}')$, and $m(\boldsymbol{x})$ is the evaluation of the mean prior. A GP is the generalization of the finite-dimensional multivariate normal distribution,

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m},\Sigma),$$

that can be read as "the vector $\boldsymbol{f}$ is a sample of a multivariate Gaussian distribution with mean $\boldsymbol{m}$ and covariance matrix Σ". In other words, the GP is a generalization of the multivariate Gaussian distribution defined over function (not vectors).

Practically speaking, when defining a prior Gaussian distribution, we must define the mean vector and the covariance matrix, while for a prior GP, we have to define the mean function[1] and the kernel function, which will be used to build the covariance matrix later on when creating a multivariate Gaussian distribution out of a GP to take into account that we only have a finite amount of data. The most popular and generic kernel is the Squared Exponential (SE), i.e.,

$$k_{SE}(x_i,x_j) = \sigma_p^2 \exp\left(-\frac{1}{2}\left(\frac{(x_i - x_j)^2}{\ell^2}\right)\right),$$

where x_i and x_j are pairs of data points in the input space. The kernel hyperparameters, σ_p^2 and ℓ are, respectively, the prior uncertainty (or variance) and the horizontal lengthscale. The choice of the hyperparameters will be highlighted in Sec. 3.1.4.

Fig. 3.1 shows the prior distribution of the function values f for different input locations x. The shaded area shows the uncertainty of the function value at a certain input position. The colored outputs are the samples drawn from the GP model. The prior distribution has a covariance matrix Σ [2] with non-zero extra diagonal terms. When a multivariate Gaussian distribution has a diagonal covariance matrix, it means that all the elements of the distributions are independent; sampling one time from a n-dimensional diagonal multivariate Gaussian will generate n independent samples, that could have been sampled independently: we are sampling noise. On the other hand, when the elements on the extra diagonal terms are non-zero, it means that the different values of the function are correlated, resulting in samples that look smooth, as in Fig. 3.1. Each of the colored functions is a sample drawn from the GP prior.

3.1.2 Posterior

To compute the posterior, after observing some data in Fig. 3.2, we need to define a likelihood function that captures the probability of observing the data given the function values at specific points. The primary reason is that we typically cannot directly observe the data from the real function; instead, we only have access to samples that are corrupted by measurement noise, i.e.,

$$\boldsymbol{y} = f(\boldsymbol{X}) + \epsilon$$

where

$$\epsilon \sim \mathcal{N}(0,\sigma_n^2).$$

[1] In the absence of any prior knowledge, the mean function is usually set to zero.

[2] built using the kernel

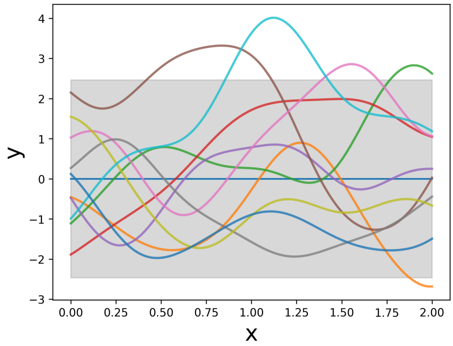


Figure 3.1: Prior distribution and samples were drawn from it. The covariance matrix has non-zero extra diagonal terms.

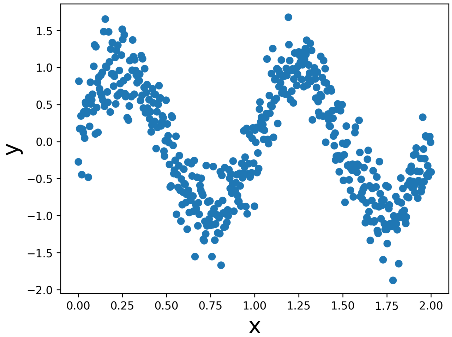


Figure 3.2: Toy example: noisy data given as labels to our model.

and $\boldsymbol{X}$ and are the observed (or measured) input and output of the function that we want to model. Hence, the likelihood function can also be assumed to be Gaussian, i.e.

$$p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{X}) \sim \mathcal{N}(f(\boldsymbol{X}),\sigma_n^2\boldsymbol{I}).$$

Differently from the GP prior, in this case, we are actually assuming that the measurements $\boldsymbol{y}$ are going to be affected by noise, which is why the likelihood function $p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{X})$ is model as a Gaussian that has mean f and as covariance matrix the identity times the squared *likelihood noise*, σ_n^2.

The posterior combines the prior and the likelihood, using Bayes theorem, i.e.,

$$\overbrace{p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})}^{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{X})}^{\text{likelihood}}\overbrace{p(\boldsymbol{f}|\boldsymbol{X})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y}|\boldsymbol{X})}_{\text{marginal likelihood}}} = \mathcal{N}(\boldsymbol{\mu}_f,\Sigma_f),$$

where the resulting posterior is still a Gaussian distribution.

Given the calculation of the posterior distribution, the prediction of new output values can be computed as:

$$p(\boldsymbol{f}_*|\boldsymbol{y},\boldsymbol{X}) = \int p(\boldsymbol{f}_*|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})d\boldsymbol{f}, \tag{3.1}$$

where we are marginalizing over the posterior distribution of the function, i.e., $p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})$, given the conditioned prior model, i.e. $p(\boldsymbol{f}_*|\boldsymbol{f})$.

Considering that $p(\boldsymbol{f}_*,\boldsymbol{f})$ is a Multivariate Gaussian Distribution by definition, then $p(\boldsymbol{f}_*|\boldsymbol{f})$ can be obtained by conditioning with respect to $\boldsymbol{f}$. The distribution of $\boldsymbol{f}$ can be computed with the posterior distribution given the evidence of the data. Given that all the terms of the integral are Gaussians, the integral is also Gaussian. So, if $p(\boldsymbol{f}_*|\boldsymbol{y},\boldsymbol{X})$ is Gaussian then $p(\boldsymbol{f}_*,\boldsymbol{y}|\boldsymbol{X})$ is a multivariate Gaussian distribution defined as

$$\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{f}_*\end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0},\begin{bmatrix}K(\boldsymbol{X},\boldsymbol{X})+\sigma_n^2\boldsymbol{I} & K(\boldsymbol{X},\boldsymbol{X}_*)\\ K(\boldsymbol{X}_*,\boldsymbol{X}) & K(\boldsymbol{X}_*,\boldsymbol{X}_*)\end{bmatrix}\right).$$

Hence, to make predictions, the mean and the variance of the posterior distribution can be computed as the conditional mean and variance of the multivariate Gaussian distribution, according to

$$\boldsymbol{\mu}_{f_*} = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y} \tag{3.2}$$

$$\Sigma_{f_*} = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*) \tag{3.3}$$

while the variance of the measured signal also has the likelihood noise added on the diagonal, i.e.,

$$\Sigma_{y_*} = \Sigma_{f_*} + \sigma_n^2 \boldsymbol{I}$$

where $\boldsymbol{X}_*$ are the prediction inputs and $\boldsymbol{X}$ and $\boldsymbol{y}$ are the training input and outputs and the correlations $\boldsymbol{K}$ are computed using a kernel function. It is worth mentioning that the prediction only relies on the recorded $\boldsymbol{X}$ and $\boldsymbol{y}$ and not on a set of parameters (like in a neural network). For this reason, GPs are usually denoted as a *non-parametric* method.

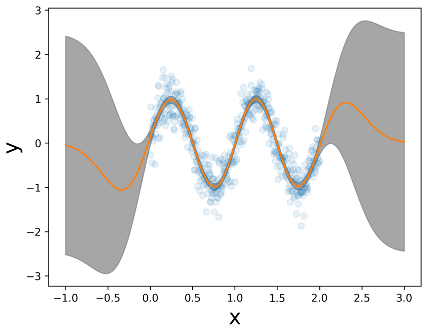


Figure 3.3: Posterior distribution of f.

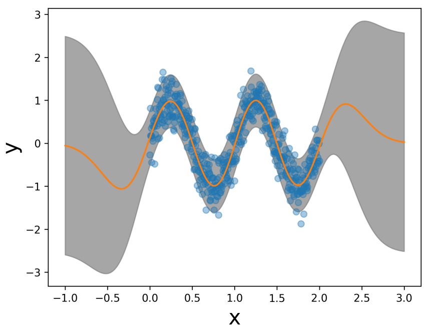


Figure 3.4: Posterior distribution of y.

Fig. 3.3 and 3.4 show the prediction of the GP for a range of input that goes beyond the training set. It is possible to notice that the mean converges to the zero prior when far away from the data. In Fig. 3.4, beyond capturing the *epistemic* uncertainty (lack of knowledge), the prediction also captures the *aleatoric* uncertainty (due to the sensor noise).

3.1.3 Extrapolation-free prediction †

In the Chapters 5 and 6, GPs are going to be used to control the position and orientation attractor of robot manipulators. However, the properties of convergence to the prior is not always desired. Let us assume that, for example, we are learning the desired end-effector orientation as a function of the current robot position. If the robot is dragged far away from the data distribution, the prediction can converge to the mean, i.e. zero, which may lead the robot to an undesired orientation. This is why an alternative inference rule is proposed to avoid the convergence back to the prior mean but without losing the epistemic uncertainty quantification. For example, we can define the mean prediction always as the most correlated point of the mean posterior $\mathbb{E}[f]$, i.e.,

$$\boldsymbol{\mu}_{f_*} = \hat{\boldsymbol{K}}(\boldsymbol{X}_*, \boldsymbol{X})\mathbb{E}[f]$$

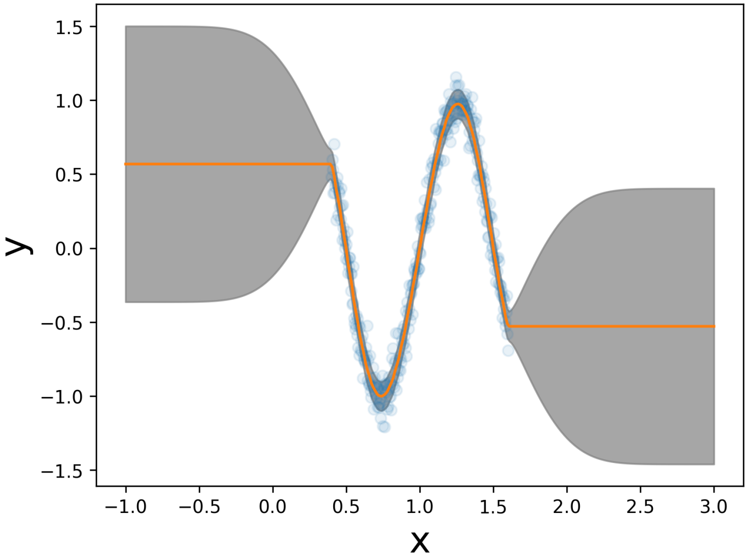


Figure 3.5: Extrapolation-free prediction. The GP mean prediction does not converge back to the zero mean of the model but predicts the posterior of the most correlated point in the dataset. On the other hand, the uncertainty is still calibrated, increasing to the prior uncertainty.

where the operator "$\hat{K}$" returns a zero-matrix with ones located on the location of the maximum value along each row. In Ch. 6, a similar formulation will be obtained to set the attractor for the robot to the most correlated point on a trajectory with a space-time encoding.

Fig. 3.5 depicts how the new prediction would not converge to the zero mean but predicts the most correlated value of the mean of the posterior $p(\boldsymbol{f}|\boldsymbol{y})$ while retaining a calibrated uncertainty quantification.

3.1.4 Hyperparameters Optimization

The GP hyperparameters, e.g., the prior uncertainty, the horizontal lenghtscales, and the likelihood noise, are the only variables to be tuned when fitting a GP.

Differently from the least square estimation, which tries to minimize the residuals on the output prediction, we rather try to find the hyperparameters that would maximize the probability of sampling our data from the prior distribution. To compute the probability of the data we use the marginal likelihood that is the integral along all the possible output functions sampled from our prior distribution, i.e.

$$p(\boldsymbol{y}|X) = \int p(\boldsymbol{y}|\boldsymbol{f}, X)p(\boldsymbol{f}|X)d\boldsymbol{f}.$$

This is also known as the evidence of the data given the model. This measure of evidence is crucial for the optimization of the hyperparameters of the kernel function because it allows finding the set of hyperparameters that would better fit the data, for example, detecting the right energy of the function (prior uncertainty), the right smoothness (horizontal lengthscale) and the sensor (or likelihood) noise.

When dealing with Gaussian prior and Gaussian likelihood, the marginal likelihood can be computed analytically and used to optimize the hyperparameters of the kernel function. Given the exponential nature of the Gaussian distribution, the (natural) logarithm of the marginal likelihood can be obtained as the sum,

$$\log(p(\boldsymbol{y}|\boldsymbol{X})) = \underbrace{-\frac{1}{2}(\boldsymbol{y}^{\top}\boldsymbol{K}_y^{-1}\boldsymbol{y})-}_{\text{data fit cost}} \quad \underbrace{\frac{1}{2}\log|\boldsymbol{K}_y|}_{\text{complexity term cost}} \quad -\frac{n}{2}\log(2\pi) \tag{3.4}$$

where $\boldsymbol{K}_y = \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2\boldsymbol{I}$ and n is the number of data points.

It is worth mentioning that the inversion of the covariance matrix in Eq. (3.4) scales with a computational cost of $\mathcal{O}(n^3)$, where n is the number of data points and needs to be computed at each timestep of the parameter optimization. This becomes prohibitive when we have a big dataset of the order of hundreds of thousands of data points. Nevertheless, some approximation techniques can be used to reduce the computational cost; see Section 3.4. Moreover, the inference of the optimal lengthscales through the maximization of the likelihood can also be used to automatically find the relevance of each of the input features.

3.1.5 Automatic Relevance Determination

The input values of the kernel can also have more than one feature vector, i.e. positions in x,y, and z of a robot, and different horizontal lengthscales can be used in each of the input features to scale the different distances in each dimension. This technique is called Automatic Relevance Determination (ARD). The squared exponential kernel with ARD active looks like this,

$$k(x_i, x_j) = \sigma_p^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{(x_{i,d}-x_{j,d})^2}{\ell_d^2}\right)\right) \tag{3.5}$$

where D is the number of features of the inputs, x_i and x_j are data points in the input space and σ_p^2 and ℓ_d are, respectively, the prior uncertainty and the horizontal lengthscale.

When the kernel has a different lengthscale for each of the input features, and we optimize the kernel using the marginal likelihood maximization, the optimization process automatically determines the relevance of features in a model, meaning it decides which features are important for predicting the output variable and which can be safely ignored. The optimization process is always trying to find the least complex model (Occam's Razor); hence, having one of the lengthscales as big as possible decreases the complexity term without affecting the data fit term. This enables us to automatically identify which features of the input are irrelevant for making predictions. When a kernel exhibits different lengthscales across various features, we refer to this as the Automatic Relevance Determination (ARD) feature being activated.

When learning robot policies, like in Ch. 4, 5 and 6, that may have redundant features or features with different units of measure, e.g. Cartesian position and orientation, we must infer different values to scale different features. This ensures that in kernel computation, we are not "mixing apples with oranges" and each feature is normalized by a different

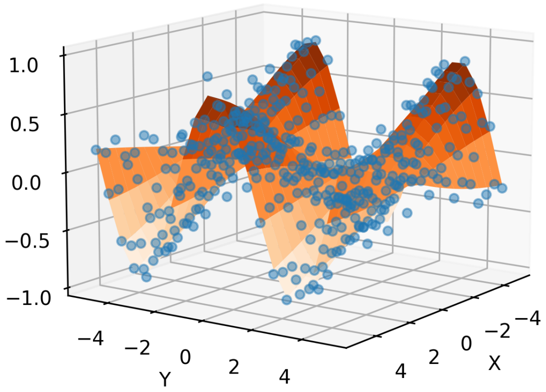


Figure 3.6: Automatic Relevance Determination example. The blue points represent the data, and the orange represents the fitted function. The function is smoother in the x direction than the y direction. For this reason, the hyperparameter tuning converges to two different lengthscales in the two input directions: the horizontal lengthscale in x is 7.91, while the horizontal lengthscale in y is 1.91. A larger lengthscale means that the process identifies the function as smoother in that direction.

lengthscale with the optimal value and right unit of measure. Fig. 3.6 shows the fitting of the data that are sampled from a function that has less variability in x than in y. The GP converged to two different lengthscales, the one in x that is significantly larger than the one in y.

3.2 INCREMENTAL LEARNING WITH GAUSSIAN PROCESS REGRESSION†

Incremental learning is a machine learning paradigm where the model is updated incrementally over time as new data becomes available. This technique is particularly crucial when we lack all the data initially or, more importantly, when the underlying true function is evolving or changing over time. For example, in Ch. 4 and 5, the robot policy is initialized with a kinesthetic demonstration, but more data are aggregated online to adjust the behavior to apply a larger force when performing force interaction tasks or to go faster when learning to pick at non zero velocity.

The incremental learning process can be divided into two phases: the first phase is the training phase, where the model is trained on a set of data, and the second phase is the incremental phase, where the model is updated with new data. The incremental phase can be further divided into two steps: the first step is the prediction step, where the (old) model is used to predict the output on a test point, and the second step is the update step, where the model is corrected if new labels are provided. The incremental learning process can be used to train a model on a large dataset and then update the model with new data as it becomes available. Active learning is a special case of incremental learning in which a learning algorithm can interactively query a human user (or some other information source) to label new data points with the desired outputs.

GPs are the perfect candidate to perform active learning given the estimation of the output uncertainty. Imagine wanting to fit a model but labeling the least amount of points to keep the model light and the "supervisor" less busy. The idea is to make predictions on many points of the workspace, such as on a grid, and then only label where the prediction is the most uncertain. This is also the principle behind Bayesian Optimization.

3.2.1 UNCERTAINTY-AWARE DATA AGGREGATION

If we consider having enough data to infer the hyperparameters of the kernel already, then, when aggregating more data to the model, only the update of the covariance matrix is performed while the hyperparameters are kept fixed. Moreover, performing active learning in the classical interpretation is unsafe when performing it on a robot manipulator. Imagine if the robot deliberately decides on which part of the Cartesian space it has to move to ask the user the best action to take there. This may induce stress in the user and eventual dangerous behaviors. For this reason, this thesis proposes an uncertainty-aware data aggregation that is based on the same principle of active learning, where the exploration is led by policy mean behavior and user corrections. In fact, the robot will ask to provide a label if the uncertainty prediction of Eq. (3.3) is bigger than a certain percentage of the prior uncertainty, i.e.

$$\sigma^2_{f_*} > \beta \sigma^2_p,$$

where β is the threshold, e.g. $\beta = 0.2$. Moreover, in a robotics scenario, where we may explore the workspace to regress a certain function, this uncertainty-aware data aggregation and update of the covariance matrix allows us to aggregate data to make a confident prediction the next time that point is visited. Given the uncertainty awareness on the prediction, the aggregation is not performed when the uncertainty is little enough or no label is provided by the teacher.

To avoid recomputing the inverse of the covariance matrix from scratch, a least square update of the model can be performed [167]. Given the update covariance matrix,

$$\boldsymbol{K}_{n+1} = \begin{bmatrix} \boldsymbol{K}_n(\boldsymbol{X},\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_{new}) \\ \boldsymbol{K}(\boldsymbol{X}_{new},\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_{new},\boldsymbol{X}_{new}) \end{bmatrix}$$

the updated inverse covariance matrix becomes,

$$\boldsymbol{K}^{-1}_{n+1} = \begin{bmatrix} \boldsymbol{K}_n(\boldsymbol{X},\boldsymbol{X})^{-1} + g\boldsymbol{e}\boldsymbol{e}^\top & -g\boldsymbol{e} \\ -g\boldsymbol{e} & g \end{bmatrix},$$

where $\boldsymbol{e} = \boldsymbol{K}_n(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_{new})$ and $g = (\boldsymbol{K}(\boldsymbol{X}_{new},\boldsymbol{X}_{new}) - \boldsymbol{K}(\boldsymbol{X}_{new},\boldsymbol{X})\boldsymbol{e})^{-1}$.

Although this uncertainty-aware aggregation rule queries and aggregates data only when the model is uncertain, this does not allow for the modification of the prediction in regions with low uncertainties. In the context of robot manipulation, where the robot has to learn to go faster or push harder in certain regions of space, we must also define a way to update the model where the provided label does not match the prediction.

3.2.2 INTERACTIVE MODEL UPDATE

One of the main contributions of this thesis is the formalization and the use of online updates of robot policies that are learned with GPs. However, when dealing with real data that can be scarce and noisy, the update can be challenging, leading to overfitting or intangible changes in the prediction. Nevertheless, the probabilistic and non-parametric nature of the GPs allows for a zero-shoot update of the model without having to rely on a gradient descent optimization procedure.

We will introduce two update rules and validate them on a toy example of the update of a sine and a noisy sine with new sampled data also from a sine but with a larger amplitude. We will also compare the update behavior with other update rules proposed in the literature.

Fig. 3.7 depicts the original data and the learned posterior distribution, but also the new data that are generated from a modification of the underlying function. Fig. 3.8 shows the same example but with the data that are sampled from a noisy signal. We are going to use these two examples to show the behavior of two update rules that are used throughout the development of this thesis and the advantages and disadvantages of each of them.

Given the old labels, i.e. $\boldsymbol{y}_{old}$ and knowing the kernel function and the likelihood noise, we can define the distribution from where the updated data of the model and the observed data, i.e. $\boldsymbol{y}_{corr}$ are sampled from,

$$\begin{bmatrix} \boldsymbol{y}_{new} \\ \boldsymbol{y}_{corr} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y}_{old} \\ \boldsymbol{\mu}(\boldsymbol{X}_*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*) \\ \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*) + \sigma_{n,corr}^2 \boldsymbol{I} \end{bmatrix} \right), \tag{3.6}$$

where $\boldsymbol{\mu}(\boldsymbol{X}_*) = \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y}_{old}$ is the mean prediction corresponding to the new recorded points, $\boldsymbol{X}_*$, using the old posterior distribution.

Pseudoinverse update rule The new prior distribution has the kernel function that is the same as the original prior, but the mean distribution is the previous data, and the previous prediction of the model relies only on the data before the update. The noise of the correction labels $\sigma_{n,corr}^2$ can be different than the one of the original data σ_n^2. However, in the lack of extra information, it will be set equal to the one of the data. The first update rule that was proposed for the first time in [55] relies on the update of $\boldsymbol{y}_{old}$ such that to maximize the likelihood of observing the new data $\boldsymbol{y}_{corr}$. The Gaussian likelihood is maximized when the prediction coincides with the distribution, i.e.,

$$\mathbb{E}(p(\boldsymbol{y}_{corr}|\boldsymbol{y}_{new})) = \boldsymbol{\mu}(\boldsymbol{X}_*) + \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1}(\boldsymbol{y}_{new} - \boldsymbol{y}_{old}) := \boldsymbol{y}_{corr},$$

bringing us to the following update rule,

$$\boldsymbol{y}_{new} = \boldsymbol{y}_{old} + \boldsymbol{A}^{+}(\boldsymbol{y}_{corr} - \boldsymbol{\mu}(\boldsymbol{X}_*)), \tag{3.7}$$

where $\boldsymbol{A} = \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1}$ and $^+$ denoted the pseudo-inverse, considering that the the $\boldsymbol{A}$ matrix is not necessarily squared. We denote this update rule as the *pseudoinverse update*.

In Fig. 3.9, we can appreciate how the updated model prediction, which relies on the new updated $\boldsymbol{y}_{new}$, perfectly adapted to match the new labels both in regions where there were already labels and also where not. However, from the update rule of Eq. (3.7), we can notice that we are not taking into account the eventual likelihood noise $\sigma_{n,corr}^2$ of the new labels. Although the hypothesis of noise-free data can hold in some circumstances when it does not, it may result in an unstable model update, as depicted in Fig. 3.10. This motivates us to find an alternative formulation for the update rule.

Conditioning update rule The generative joint distribution of the updated labels and the new sampled labels of Eq. (3.6) can be directly conditioned to compute the updated values of $\boldsymbol{y}_{new}$, i.e.,

$$\mathbb{E}(p(\boldsymbol{y}_{new}|\boldsymbol{y}_{corr})) = \boldsymbol{y}_{old} + \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)(\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*) + \sigma^2_{n,corr}\boldsymbol{I})^{-1}(\boldsymbol{y}_{corr} - \boldsymbol{\mu}(\boldsymbol{X}_*)). \tag{3.8}$$

Although this rule seems very similar to the previous rule, it is a complete paradigm shift. We are not trying to condition the distribution and change the labels such that we are matching the output of the new distribution with the new observed labels, but we directly infer what should be the new data distribution, $\boldsymbol{y}_{new}$, by conditioning the joint distribution on the new labels $\boldsymbol{y}_{corr}$. For this reason, we denote this as the *conditioning update rule*.

The quantification of the noise in Eq. (3.8), i.e., $\sigma^2_{n,corr}\boldsymbol{I}$, also has the advantage of being robust to the uncertainties in the provided *correction labels*. Fig. 3.11 and 3.12 depict the behavior of the update and show robust updates also when there is a likelihood noise on the original and the new labels. Additionally, to increase the robustness of wrong labels provided during the update, a variable $\sigma^2_{n,corr}$ can be adopted according to the source of the new corrective label, for example, one expert user may be less likely to provide wrong labels, i.e. the noise $\sigma^2_{n,corr}$ would be zero while in case of a novice user, the noise can be set to be higher to avoid to update the model too quickly to possibly wrong labels. Moreover, the multiplication by $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)$ in the update rule of Eq. (3.8) has another powerful meaning: if the new label is not correlated with old labels, because they are located in originally uncertain regions, the update rule will have not effect on the labels. This is evident in both Figs. 3.11 and 3.12 where the new function prediction does not match the new function in the uncertain region on the right. This is desirable since we only update the labels that are correlated with the corrective labels and, in uncertain regions, perform the uncertainty-aware aggregation as depicted in Figs 3.13 and 3.14.

The careful reader may question why the update rule is not used also to update the uncertainty of the labels, resulting in a heteroscedastic model. This is an interesting research direction that was, however, out of scope in this thesis; hence, the proposed update rules are only used to update previously recorded data using the received corrective input.

Swapping data rule Having a well-performing update rule for the model that is updated online is fundamental for effective teaching, and the previous two update rules highlighted the possibility of considering the correlation of the new labels with the elements in the dataset and how to effectively perform the update. These update rules differ from other update rules for interactive learning with GPs [174] or online model learning [121] that only consider an independent swapping data rule. Their approach consists of finding the (only) most correlated point (if the uncertainty prediction is below a threshold) and swapping the only label with the new one in the dataset. This approach does not consider the effect of possible correlation with more data in the update of the model. Figs. 3.15 and 3.16 show how, since only a few labels are moved on the new observed data points, the prediction, close to the location of the old data, still converges to the old function, since the greatest part of the label was not updated correctly.

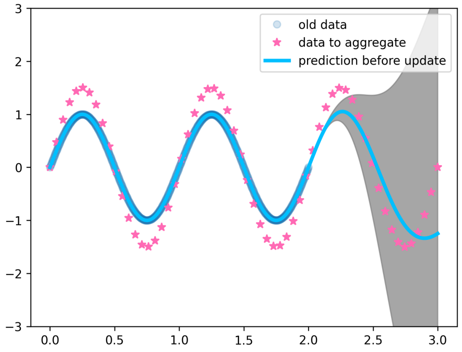


Figure 3.7: Noise-free data, old posterior prediction, and new noise-free data to aggregate.

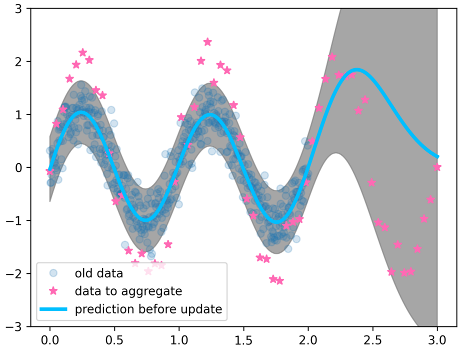


Figure 3.8: Noisy data, old posterior prediction, and new noisy data to aggregate.

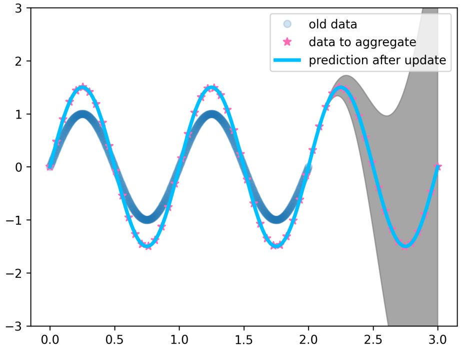


Figure 3.9: Pseudoinverse update rule (noise-free data).

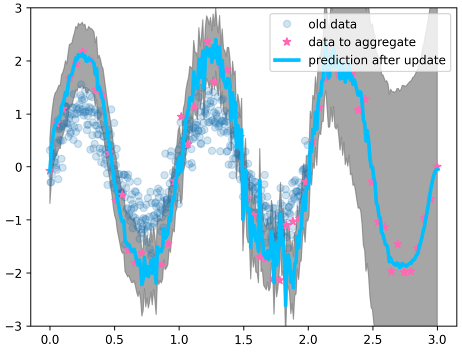


Figure 3.10: Pseudoinvere update rule (noisy data).

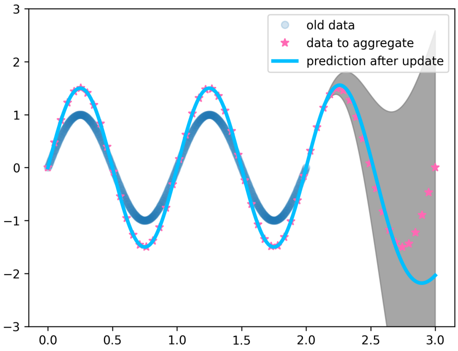


Figure 3.11: Conditioning update rule (noise-free data).

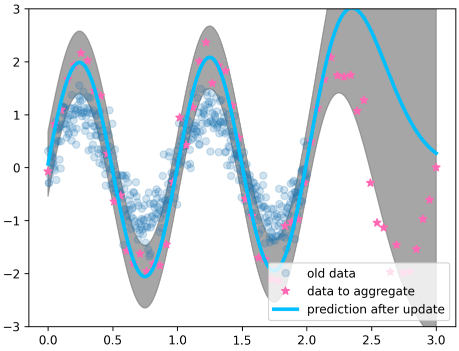


Figure 3.12: Conditioning update rule (noisy data).

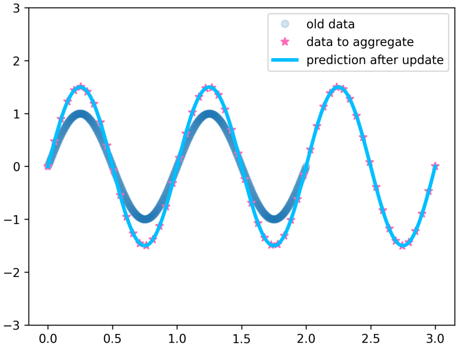


Figure 3.13: Conditioning update rule plus uncertainty-aware aggregation (noise-free data).

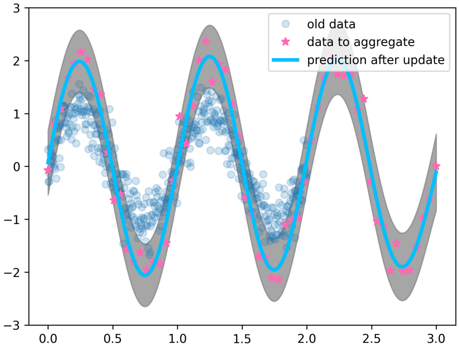


Figure 3.14: Conditioning update rule plus uncertainty-aware aggregation (noisy data).

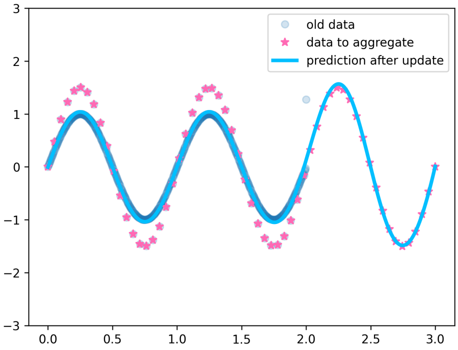


Figure 3.15: Swapping data plus uncertainty-aware aggregation plus uncertainty-aware aggregation (noise-free data).

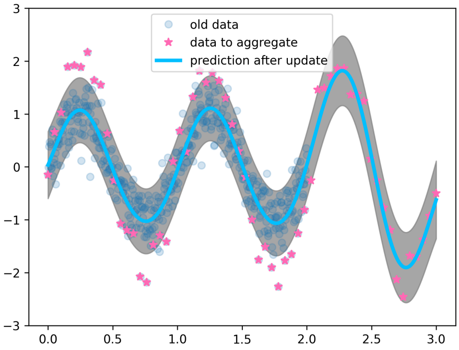


Figure 3.16: Swapping data plus uncertainty-aware aggregation (noisy data).

3.3 DERIVATIVES OF GAUSSIAN PROCESSES

In Ch. 7, GPs will be used to learn a transformation to generalize the learned robot policy and make use of the derivative of the non-linear transformations with respect to the input. Fortunately, the derivative of a GP is also a GP, and its existence will depend on the differentiability of the mean function and the kernel function. The correlation between derivative samples can be expressed as the double partial derivative $k_{11} = \frac{\partial}{\partial x_i \partial x_j} k(x_i, x_j)$ while the correlation between derivative samples and function samples is $k_{10} = \frac{\partial}{\partial x_i} k(x_i, x_j)$. The observed data, $\boldsymbol{y}$, and the derivative of the posterior distribution $\boldsymbol{f}'_*$ are jointly Gaussian distributed as,

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}'_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{00}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K}_{01}(\boldsymbol{X},\boldsymbol{X}_*) \\ \boldsymbol{K}_{10}(\boldsymbol{X}_*,\boldsymbol{X}) & \boldsymbol{K}_{11}(\boldsymbol{X}_*,\boldsymbol{X}_*) \end{bmatrix}\right),$$

where the type of kernel derivative is specified at the feet of each kernel matrix.

Thus, the mean and variance prediction of the derivative of the GP can be formulated as:

$$\begin{aligned} \mathbb{E}[f'_*(x)] &= \boldsymbol{\mu}' = \boldsymbol{K}_{10}(\boldsymbol{X}_*,\boldsymbol{X})(\boldsymbol{K}_{00}(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y} \\ \mathbb{E}[f'_*(x) f'_*(x)] &= \Sigma' = \boldsymbol{K}_{11}(\boldsymbol{X}_*,\boldsymbol{X}_*) - \boldsymbol{K}_{10}(\boldsymbol{X}_*,\boldsymbol{X})(\boldsymbol{K}_{00} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{K}_{01}(\boldsymbol{X},\boldsymbol{X}_*). \end{aligned} \tag{3.9}$$

Fig 3.17 shows the original noisy sine, the prediction of the model, and the prediction of the derivatives. It is worth underlying that the function is not obtained by differentiating the mean prediction of the model, numerically or automatically, but the result is obtained in a fully Bayesian way that also estimates the uncertainties on the derivative prediction, according to Eq. (3.9). Uncertainty quantification of a stochastic function's derivative becomes essential for propagating uncertainties associated with the function's derivative.

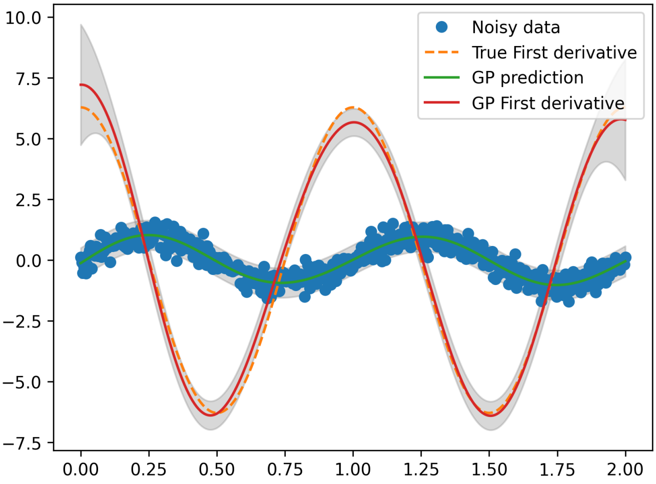


Figure 3.17: Derivative of a Gaussian Process is a Gaussian Process. The figure depicts the original data, the GP posterior prediction with uncertainty, and the prediction of the derivative of the GP mean and uncertainty that is obtained by differentiating the GP model.

3.3.1 Uncertainty rate quantification †

Another contribution of this thesis, that will be highlighted in Ch. 4 and Ch. 5, is the quantification and the rejection of the estimated model uncertainty. For example, a robot that learns the desired Cartesian velocity as a function of its position in space may end up in regions of space that have high uncertainties due to the lack of data. However, thanks to the closed-form estimation of uncertainties, it is possible to find the direction of maximum growth of the uncertainties and push the robot in opposite directions. Analytically, the uncertainty rate (as a function of the model input) can be quantified as,

$$\frac{\partial \Sigma}{\partial \boldsymbol{x}_*} = \boldsymbol{K}_{10}(\boldsymbol{x}_*, \boldsymbol{x}_*) - 2\boldsymbol{K}_{10}(\boldsymbol{x}_*, \boldsymbol{X})(\boldsymbol{K}_{00} + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{K}_{00}(\boldsymbol{X}, \boldsymbol{x}_*). \tag{3.10}$$

The derivative of the variance (with respect to the input variable) is not the variance of the derivative function. As a matter of fact, for regions that are infinitely far away from the data, the variance of the GP converges to the prior uncertainty; therefore, the variance derivative with respect to the input is zero, i.e. the uncertainty is constant. On the other hand, the variance of the derivative function must converge to the prior uncertainty of the derivative function, i.e. $\boldsymbol{K}_{11}(\boldsymbol{X}_*, \boldsymbol{X}_*)$ which cannot be zero.

3.4 Approximate Gaussian Process Regression with Variational Inference

When dealing with many data points in the order of thousands, for example, when collecting many demonstrations from a human teacher, the computational cost of the inversion of the covariance matrix becomes prohibitive, particularly when performing the estimation of the kernel hyperparameters and the likelihood noise where the inversion needs to be computed for each optimization step. Variational inference [159] aims to approximate the true posterior $p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y})$ with a simpler distribution $q(\boldsymbol{f})$ defined as:

$$q(\boldsymbol{f}) := \int p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u}, \tag{3.11}$$

where

$$q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{S}).$$

The variational distribution $q(\boldsymbol{u})$ relies on a set of inducing outputs, i.e. $\boldsymbol{u}$, that corresponds to a set of inducing inputs $\boldsymbol{Z}$. The inducing outputs are Gaussian distributed with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{S}$. For a thorough review of approximation methods on GP, see [132].

In [71], a novel lower bound to the true marginal log-likelihood (ELBO) is proposed to fit the variational distribution by maximizing the likelihood of the data given as labels, i.e.,

$$\begin{aligned}\log(p(\boldsymbol{y}|\boldsymbol{X})) \geq \sum_{i=1}^{n} \Bigg\{ & \log \mathcal{N}(y_i | \boldsymbol{K}(x_i, Z)\boldsymbol{K}(Z,Z)^{-1}\boldsymbol{m}, \sigma_n^2) + \\ & -\frac{1}{2}\sigma_n^{-2}\big(\boldsymbol{K}(x_i,x_i) - \boldsymbol{K}(x_i,Z)\boldsymbol{K}(Z,Z)^{-1}\boldsymbol{K}(Z,x_i)\big) + \\ & -\frac{1}{2}\mathrm{tr}\big(\sigma_n^2 \boldsymbol{S}\boldsymbol{K}(Z,Z)^{-1}\boldsymbol{K}(Z,x_i)\boldsymbol{K}(x_i,Z)\boldsymbol{K}(Z,Z)^{-1}\big)\Bigg\} + \\ & -\mathrm{KL}(q(\boldsymbol{u})\|p(\boldsymbol{u})).\end{aligned}$$

The key property of this new lower bound of the true marginal log-likelihood is that it can be written as a sum of terms. This implies that when we are searching for the optimal inducing inputs, $\boldsymbol{Z}$, and variational parameter, $\boldsymbol{m}$ and $\boldsymbol{S}$, we can just sum over a (small) batch of our dataset, so the computational cost is only proportional to the batch size we are using in the update of the parameters. However, it is worth noticing that we still need to compute the inverse of the covariance evaluated on the inducing points, this has a computational cost of $\mathcal{O}(m^3)$. So, it only scales with the number of inducing points, not the data.

The fact that we are still able to maximize the (lower bound of the) marginal log-likelihood of the data without having to use the whole set of labels at every iteration opens the door to fit GPs on (very) big datasets without losing the probabilistic interpretation of the inference of the function and the inference of the prediction.

To make predictions on a test point and using the definition of the approximate posterior, see Eq. (3.11) and the prediction integral for a standard GP, see Eq. (3.1)[3], the predictive distribution can be computed as:

$$\begin{aligned} p(\boldsymbol{f}_*|\boldsymbol{y}) &= \int p(\boldsymbol{f}^*|\boldsymbol{f},\boldsymbol{u})p(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{y})d\boldsymbol{f}d\boldsymbol{u} \\ &\approx \int p(\boldsymbol{f}^*|\boldsymbol{f},\boldsymbol{u})p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{f}d\boldsymbol{u} \\ &= \int p(\boldsymbol{f}_*|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u}. \end{aligned}$$

The result of this integral is

$$p(\boldsymbol{f}_*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{A}\boldsymbol{m}, \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*) - \boldsymbol{A}(\boldsymbol{K}(\boldsymbol{Z},\boldsymbol{Z}) - \boldsymbol{S})\boldsymbol{A}^\top),$$

where $\boldsymbol{A} = \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{Z})\boldsymbol{K}(\boldsymbol{Z},\boldsymbol{Z})^{-1}$ and we can compute the mean and variance of a test point in $\mathcal{O}(m^2)$. As you can notice, we also do not have the inputs or the output label anywhere in the prediction. The prediction computational cost will not be affected by the dimension of the training set but only by the mean and variance of the approximated distribution $q(\boldsymbol{u})$. Fig. 3.18 depicts the prediction that only relies on five inducing points, in red, located in $\boldsymbol{Z}$ with value $\boldsymbol{m}$.

[3] Also assuming that $p(f,u|y) = p(f|u,y)p(u|y) \approx p(f|u)q(u)$, considering that f is independent from y and only depends on u. $p(u|y)$ is approximated with our variational distribution $q(u)$.

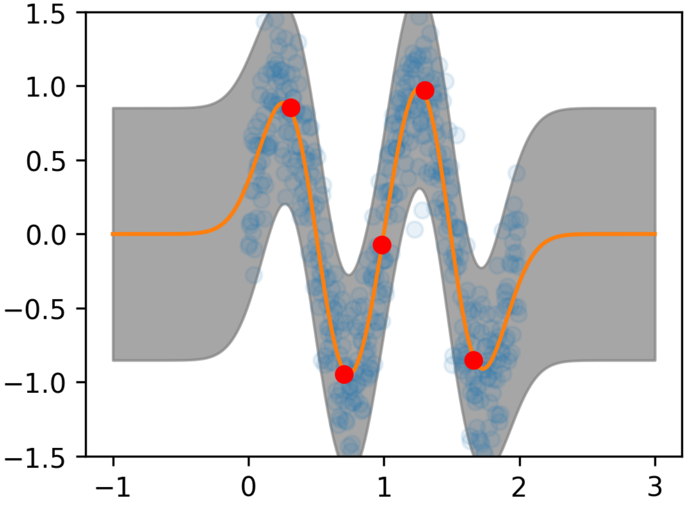


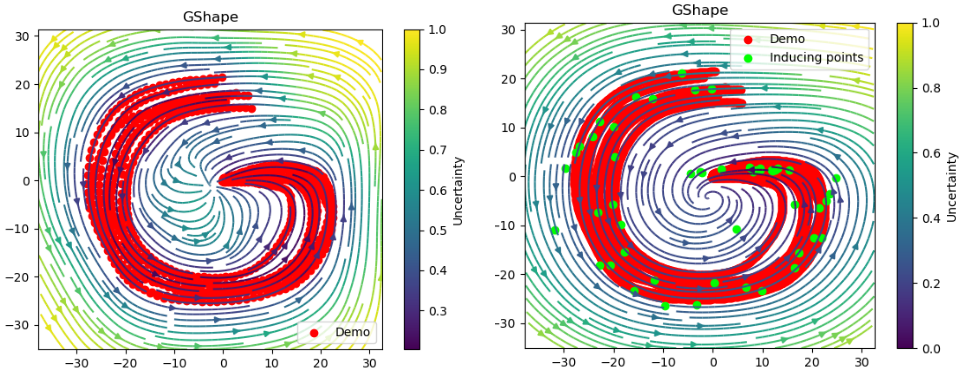
Figure 3.18: Fitting of noisy sine using Stochastic Variational Gaussian Process.

3.5 Multioutput Gaussian Process

When dealing with multiple outputs, such as when modeling the desired robot velocity, in different Cartesian directions, as a function of the Cartesian position, each of them can be modeled as an independent GP with different kernel functions and hyperparameters. However, the kernel function can also be shared between the different outputs. This is useful when the different outputs are correlated, e.g. the velocity of a robot in different Cartesian directions. When modeling multiple outputs, like velocities, it makes sense to share the horizontal lengthscales of the kernel, given that the smoothness of the motion would be independent of the Cartesian direction. On the other hand, the vertical lengthscale, or prior uncertainty, can be shared only if the energy of the two outputs is the same and they have the same units of measure. However, when different prior uncertainties are used, the maximum variance for each output will be different.

Fig. 3.19 depicts the vector field learned with a GP given multiple demonstrations of a planar motion drawing the letter G. The GP models the desired Cartesian velocity ad a function of Cartesian position, i.e. $\dot{x} = f(x)$ given the recorded labels $\dot{X}$ and X. The lengthscales are shared among the different outputs, i.e. the velocity in the x and y direction, while the prior uncertainty is not shared. The uncertainties that represent a measure of ignorance, i.e. lack of knowledge, increase when evaluating far away from the data. Similarly to the kernel hyperparameters, the inducing point location, when learning a Variational Gaussian Process, can be optimized independently for each output or shared. Having independent variational parameters for each output increases the flexibility of the model as well as the computational cost and memory requirements. So, if the number of outputs is large or to speed up computation, it is convenient to share the variational parameters among different outputs.

Fig. 3.19b shows the vector field learned from the velocity-position labels using approximate inference, illustrated in Sec. 3.4. The figure highlights how the information of many data points can be compressed in a smaller distribution located on a reduced set of inducing points that are automatically learned when maximizing the ELBO of the data likelihood.



(a) Vector Field fitted with Multi-output Exact Gaussian Process.

(b) Vector Field fitted with Multi-output Stochastic Variational Gaussian Process.

Figure 3.19: Vector field learned from the demonstration of drawing the letter G. The input is the position and the output is the velocity. The lengthscales and the inducing points are shared among different outputs. The output lengthscales are not shared.

3.6 Conclusions

Utilizing a robot capable of swiftly learning behaviors through human demonstration and refining those skills based on human feedback represents a pivotal step in bridging the divide between learning in a computer simulation and real-world manipulation. Traditional offline data-based learning falls short of capturing the desired behaviors communicated by the user demonstration. The distinctive probabilistic and non-parametric attributes of Gaussian Processes (GPs) facilitate prompt model updates solely through the data correlation computed with the kernel function, as elaborated in this chapter. This will be exploited in the next chapters to update real robot manipulation policies.

Moreover, machine learning methods often exhibit limited extrapolation capabilities, potentially leading to hazardous behaviors in unfamiliar scenarios. Thus, incorporating uncertainty quantification into learned policies, alongside a safety-enhancing prior, becomes paramount for working among humans safely. For instance, in subsequent Chapters 4 and 5, a zero-mean prior ensures the robot remains stationary when confronted with out-of-distribution situations. Additionally, uncertainty quantification proves pivotal in guiding the robot close to the original demonstration. Furthermore, Chapter 7 delves into leveraging GPs for policy adaptation to novel scenarios, such as varied object placements or surface geometries. Here, the integration of uncertainty quantification and zero-mean priors serves to minimize distortions in the original policy during adaptation while providing estimations of uncertainty in the generalization process.

4

ILoSA: Interactive Learning of Stiffness and Attractors

Teaching robots how to apply forces according to our preferences is still an open challenge that has to be tackled from multiple engineering perspectives. This chapter studies how to learn variable impedance policies where both the Cartesian stiffness and the attractor can be learned from human demonstrations and corrections with a user-friendly interface. The presented framework, named Interactive Learning of Stiffness and Attractors (ILoSA), uses GPs for policy learning, identifying regions of uncertainty and allowing interactive corrections, stiffness modulation, and active disturbance rejection. The experimental evaluation of the framework is carried out on a Franka-Emika Panda in four separate cases with unique force interaction properties: 1) pulling a plug wherein a sudden force discontinuity occurs upon successful removal of the plug, 2) pushing a box where a sustained force is required to keep the robot in motion, 3) wiping a whiteboard in which the force is applied perpendicular to the direction of movement, and 4) inserting a plug to verify the usability for precision-critical tasks in an experimental validation performed with non-expert users. The results show that the proposed framework is able to learn the desired force profiles and adapt to the user's corrections in a safe and user-friendly manner. A video of the experiments can be found here: `https://youtu.be/MAG-kFGztws`.

This chapter is based on 🗎 ***Franzese, G.***, *Mészáros, A., Peternel, L. and Kober, J., 2021, September. ILoSA: Interactive Learning of Stiffness and Attractors. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 7778-7785) IEEE. [55].*

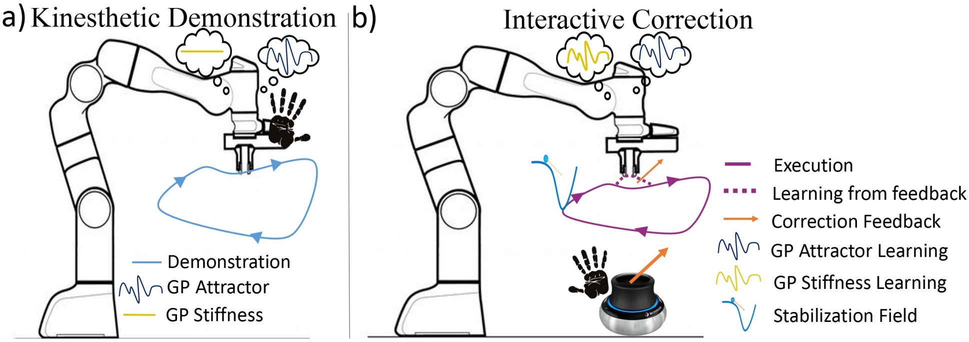


Figure 4.1: Overview of the ILoSA framework.

4.1 Introduction

Robots have long been a tool for efficiently carrying out repetitive or mundane tasks. Recently, more robotic applications have been targeted toward interacting with varying and unknown environments in order to aid people in their daily tasks. Quite often, the exact behavior required for interacting with such environments cannot be directly modeled or is simply too complex to do so. However, people already possess intuitions on how to interact with the world around them and can transfer this knowledge. In this direction, learning through demonstration has become increasingly popular for teaching robots familiar yet complex tasks in an intuitive manner [11].

Learning is especially handy for manipulation tasks, which come with the requirement of exerting a certain degree of force. The goal of a manipulation operation is not only to perform a desired trajectory but to learn the desired force that the robot has to exercise on its environment in order to accomplish the desired goal. Different methods exist for controlling the robot to perform contact tasks, from the use of force control, hybrid position-force control [133], as well as impedance control methods [73]. In addition, when robots coexist with humans, it is crucial to consider that for safe interaction, the robot should limit the force to the minimum required, as well as be compliant when interacting with elements of the environment that are not the target of the manipulation.

Out of these methods, impedance control is best suited for achieving such behavior, since force control would generate dangerous and unstable accelerations when in free movement [92], while impedance control would only converge to the nearby attractor. Furthermore, when using impedance control, for a safer and user-friendly interaction of the robot with the environment, the trajectory execution has to be performed in a feedback/reactive, not in a feed-forward, manner. This avoids the accumulation of error between the attractor and end-effector positions with the consequent generation of undesired high interaction forces and/or accelerations.

In the presented framework, both the desired attractor position and the desired stiffness are learned as a function of the robot position. This is done by using a nonlinear feedback policy that is learned from kinesthetic demonstrations and teleoperated corrections. More importantly, because the robot learns from a human demonstration, an estimation of the epistemic uncertainty of the policy is necessary for a safe execution of the trajectory. For this purpose, GP provides a non-parametric learning method that enables a good generalization in the neighborhoods of the demonstration while providing information on the confidence level of the corresponding prediction. This, in turn, can be utilized to make the robot more compliant in states the robot has not visited before, thus avoiding undesired and dangerous behaviors.

To summarize, the motivation of this chapter is to establish whether learning both attractor and stiffness policies in a reactive formulation with a GP allows the performance of manipulation tasks while ensuring a safe interaction between the robot and its environment by exploiting the information of model confidence. Additionally, we introduce a new update rule for a GP policy in order to allow data and time-efficient learning from teleoperation corrections after the initial kinesthetic demonstration and to automatically allocate the feedback as an attractor or stiffness modulation. Also, we investigate the concept of adaptive disturbance rejection with the use of a stabilization prior based on a force field proportional to the gradient of the GP variance manifold.

4.2 RELATED WORK

The impedance controller reacts proportionally to the distance from the desired reference position. This enables the robot to follow a trajectory when in free space or to apply a force when in contact with an object. This dual property is utilized within the scope of the developed framework.

In order to automatically learn policies to complete complex tasks, one of the common approaches is to apply RL algorithms such as Policy Improvement with Path Integrals (PI^2)[26, 70, 85], Natural Actor-Critic [88], and multi-optima policy search [35]. RL methods, however, tend to take a long time to achieve the desired performance level. Additionally, in order to train a policy through RL, an adequate cost function is needed for which a good understanding of the task mechanics is required.

A faster alternative is learning from demonstrations [92], which can additionally be augmented with incremental learning [128] in order to improve a demonstrated policy. Previous works have shown that while using an impedance controller it is possible to learn varying stiffness profiles in order to carry out force interaction tasks [1, 140] as well as to allow compliance in areas outside of force interaction [26].

For the purpose of learning interaction with the environment, feedback from a human operator may be needed in the form of corrections in the action space [126]. HG-DAgger[87] proposes to allow the user to take control and provide local demonstrations online which are then aggregated onto the existing database. However, the aggregation operation does not employ a data-efficient rule for updating the old database according to the current corrections.

Inspired by the exploitation of model confidence of active learning [151], we learned an overall policy that remains compliant in regions of uncertainty for the purpose of safety. GPs have proven to be a viable approach towards achieving this form of behavior in [49] where the uncertainty information was utilized in order to allocate leadership in the form of compliance to interacting agents in a bi-manual task. On the data efficiency side, Probabilistic Inference for Learning Control (PILCO) [45] successfully employed GPs in RL for learning the system model and using the information about the uncertainty for a faster search for policy optimization. Analogously, ILoSA uses the information of the model uncertainty for a data-efficient update of the policy with user corrections, for stiffness modulation in uncertain regions, and for active rejection of disturbances with a stabilization function.

In the literature, other probabilistic methods were applied for modeling the stochasticity of movement primitives [125] and inferring the desired trajectory by conditioning the model on the chosen goal. Furthermore, methods like Gaussian Mixture Models (GMMs) showed a successful application in inferring the desired manipulability ellipsoid from the consistency of multiple demonstrations [29] in combination with a geometry-aware controller [82]. Similarly, the same method was employed for the fusion of an imitation policy with a stabilization prior [130] reducing the problem of the covariate shift. However, these probabilistic approaches do not allow the interactive correction of the policy after the provided demonstration and do not design the stabilization prior as a function of the model confidence, contrary to ILoSA.

Moreover, we propose to teach the desired force through automatic inference on the increase of the attractor distance or stiffness only from user corrections. Combining this with a novel data-efficient update rule and the exploitation of the model confidence provided a user-friendly way of teaching force tasks as described in the following sections.

4.3 Framework: ILoSA

ILoSA employs two main teaching modalities: **kinesthetic demonstration** and **teleoperation feedback**, see Fig. 4.1. The first is used for initializing the policy for the desired dynamics of the end-effector. This policy is then executed in the second modality, whereby the user can provide online corrections to the policy.

The aim of the learned policy is to affect two particular aspects of the impedance control: the attractor distance $\Delta\boldsymbol{x}$, and the stiffness $\mathcal{K}$ of the end-effector. Briefly, in a Cartesian impedance control [73], the end-effector dynamics are modeled in the form of a mass-spring-damper system

$$\Lambda(\boldsymbol{q})\ddot{\boldsymbol{x}} = \mathcal{K}\Delta\boldsymbol{x} - \mathcal{D}\dot{\boldsymbol{x}} + \boldsymbol{f}_{ext},$$

where $\Lambda(\boldsymbol{q})$ is the physical system's Cartesian inertia matrix, $\mathcal{D}$ is the corresponding critical damping matrix, and $\boldsymbol{f}_{ext}$ are the external forces, see Appendix A.1 for further details on the implementation. In the proposed framework, the controlled 3-D vector $\Delta\boldsymbol{x}$ and anisotropic diagonal stiffness matrix $\mathcal{K}$ are the mean values of GPs, conditioned on the current Cartesian position $\boldsymbol{x}$ of the robot, i.e.,

$$\Delta\boldsymbol{x} = \mathcal{GP}_{\Delta\boldsymbol{x}}(\boldsymbol{x})$$

and

$$\mathcal{K} = \mathcal{GP}_{\mathcal{K}}(\boldsymbol{x})$$

where the used function approximation is a GP. In the initialization of the policy, following the kinesthetic demonstration, the hyper-parameters of the GP models are optimized for maximizing the expectation of the predicted attractor distance of the provided demonstrations. The same parameters are then used to initialize GP models of the stiffness in the three principal directions, however choosing a non-zero mean of $\mathcal{K}_{\text{mean}}$ in each direction. In case a force sensor is installed on the end-effector, the stiffness could be initialized proportionally to the recorded external force. Our goal is to show that even without a force sensor, the stiffness can be initialized to a base value, and the desired deviations can be learned with interactive human corrections.

ILoSA additionally incorporates two safety features. The first is a stabilization prior which ensures robust control. The second is a modulation function that pulls the stiffness down to zero in regions of high uncertainty. These two aspects will be explained further in the course of this section.

In the following subsections, details are reported on how the GP learns from the demonstration and corrections (Sec. 4.3.1), how the directional feedback is spread between attractor and stiffness (Sec. 4.3.2), and how a stabilization prior (Sec. 4.3.3) and stiffness modulation (Sec. 4.3.4) are obtained as a function of the process variance.

4.3.1 Interactive Learning with Gaussian Processes

Algorithm 1 Teaching Framework for ILoSA

```
**Record Kinesthetic Demonstration(s):**
while Recording Trajectory do
    $\boldsymbol{x}_t \uplus \boldsymbol{\xi}$
    $\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t$
    $\Delta\boldsymbol{x}_t \uplus \Delta\boldsymbol{x}^{demo}$
end while
Train Gaussian Processes hyperparameters
**Interactive Corrections:**
while Control Active do
    Receive($\boldsymbol{x}$)
    $[\Delta\boldsymbol{x}, \sigma^2] = \mathcal{GP}_{\Delta\boldsymbol{x}}(\boldsymbol{x})$
    $\mathcal{K} = \mathcal{GP}_{\mathcal{K}}(\boldsymbol{x})$
    if Received Human Feedback then
        $[\Delta\boldsymbol{x}_{\text{inc}}, \mathcal{K}_{\text{inc}}] \leftarrow$ **Interpret**(feedback, $\Delta\boldsymbol{x}, \mathcal{K}$)
        if $\sigma^2 > \sigma^2_{\text{Threshold}}$ then
            $\boldsymbol{x} \uplus \boldsymbol{\xi}, (\Delta\boldsymbol{x} + \Delta\boldsymbol{x}_{\text{inc}}) \uplus \Delta\boldsymbol{x}^{demo}, (\mathcal{K} + \mathcal{K}_{\text{inc}}) \uplus \mathcal{K}^{demo}$
        else
            Correct($\Delta\boldsymbol{x}_{\text{inc}} \rightarrow \Delta\boldsymbol{x}^{demo}, \mathcal{K}_{\text{inc}} \rightarrow \mathcal{K}^{demo}$)
        end if
    end if
    $\Delta\boldsymbol{x} = \mathcal{GP}_{\Delta\boldsymbol{x}}(\boldsymbol{x})$
    $\mathcal{K} = \mathcal{GP}_{\mathcal{K}}(\boldsymbol{x})$
    $\boldsymbol{f}_{\text{stable}} = -\alpha\nabla\sigma^2$
    $[\Delta\boldsymbol{x}, \mathcal{K}] =$ **Modulation**($\Delta\boldsymbol{x}, \mathcal{K}, \boldsymbol{f}_{\text{stable}}, \sigma^2$)
    Send($\Delta\boldsymbol{x}, \mathcal{K}$)
end while
```

The two equations that govern the mean and the variance of the process are

$$\mu(\boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x},\boldsymbol{\xi})(\boldsymbol{K}(\boldsymbol{\xi},\boldsymbol{\xi}) + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\xi},\boldsymbol{x})\boldsymbol{y},$$

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{K}(\boldsymbol{x},\boldsymbol{\xi})(\boldsymbol{K}(\boldsymbol{\xi},\boldsymbol{\xi}) + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{K}(\boldsymbol{\xi},\boldsymbol{x}),$$

where $k(\boldsymbol{x},\boldsymbol{x})$ is the kernel evaluated with itself in $\boldsymbol{x}$, $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\xi})$ is the covariance between $\boldsymbol{x}$ and the training inputs $\boldsymbol{\xi}$, $\boldsymbol{K}(\boldsymbol{\xi},\boldsymbol{\xi})$ is the covariance matrix of the training inputs, σ_n^2 is the likelihood noise of the training points, and $\boldsymbol{y}$ denotes the training outputs [136]. In this framework, the output can be the attractor distance $\Delta\boldsymbol{x}$ or the stiffness $\boldsymbol{\mathcal{K}}$ according to whether we are using the GP to predict one or the other during control. The used kernel is an ARD kernel described in Eq. (3.5), and since we are modeling three attractor distances, and the three principal stiffnesses, we use a multi-output GP where the horizontal length scales are shared among the different outputs while the likelihood noise and the prior uncertainty are optimized differently for Cartesian displacement and Cartesian stiffness, according to what was explained in Sec. 3.5. After the kinesthetic demonstration(s) (Fig. 4.1a), the hyper-parameters are automatically determined by marginal log-likelihood maximization, see Sec. 3.1.4.

When the interactive corrections are provided through teleoperation with the human-in-the-loop, the hyper-parameters are kept invariant because the correlation between samples (and validity of the same kernel) can also be considered invariant. The interactive correction in lines 13-21 of Alg. 1 summarizes how ILoSA exploits the use of the uncertainty measure for understanding if the robot is in a new unvisited region (line 15-16). This requires adding the corrective sample to the database. We defined this to be the uncertainty-aware data aggregation in Sec. 3.2.1. Otherwise, it determines how to spread the correction on all existing samples that are correlated with the current end-effector position without adding additional samples. The feedback is provided from a teleoperation device as a relative correction where the recorded increment is added on top of the current robot transition. So the new label to aggregate, i.e., $\boldsymbol{y}_{corr}$, is defined as the sum of the current predicted mean $\mu(\boldsymbol{x})$ and the given correction $\boldsymbol{\epsilon}$. Since the absence of likelihood noise in the teleoperated correction, we can use the *pseudoinverse update rule*, introduced in Sec. 3.2.2.

Thus, the update rule of the database (line 18) is

$$\boldsymbol{y}_{\text{new}} = \boldsymbol{y} + \boldsymbol{A}(\boldsymbol{\xi},\boldsymbol{x})^{+}\boldsymbol{\epsilon},$$

where $\boldsymbol{A}(\boldsymbol{\xi},\boldsymbol{x})^{+}$ is the pseudoinverse of $\boldsymbol{A}$ and $\boldsymbol{\epsilon}$ is the correction provided at $\boldsymbol{x}$. This rule was applied for correcting the attractor distance, stiffness, or both according to the interpretation of the human feedback, see Sec. 4.3.2.

This way of spreading corrections on the database was shown to be more user-friendly than only performing aggregation, as well as time and data-efficient. As also in other research works [20], having contradictory, incremental, or multimodal data can generate a bias of the predicted solution towards the more frequent samples. When doing policy correction, this can result in the unobservability of the effect of feedback and in the user's frustration. The proposed update rule resolves this problem and proved to also be effective in rapidly adjusting mistaken corrections provided in the previous policy roll-outs without the accumulation of old labels in the database.

4.3.2 Directional Feedback Interpretation

Our aim is to make teaching as simple as possible for non-expert users without any knowledge about robot control. As the name of the framework suggests, the goal is to learn the attractor and stiffness for the robot end-effector. Related works like [2] and [127] already investigated how to teleoperate the robot while also teaching the desired stiffness ellipsoid of the end-effector or recent works like [117] infers the modulation as a function of external forces. Contrary to this, our goal is to be able to infer the stiffness not by explicitly labeling it but through the same teleoperated feedback intended for correcting the direction of movement without the use of any expensive device and as a function of the robot's position. Accordingly, the main idea is to enable the user to incrementally correct the dynamics of the end-effector. The attractor distance increment Δx_{inc} is obtained as the teleoperated input *feedback*; however, if the resulting attractor corrections go beyond a certain limit, we instead increase the stiffness to match the ask increment of force in that direction, i.e. if $|\Delta x + \Delta x_{\text{inc}}| \geq \Delta x_{\text{lim}}$ or $\mathcal{K}_{\text{x}} > \mathcal{K}_{\text{mean}}$, the stiffness change $\mathcal{K}_{\text{x,inc}}$ is obtained by solving the equation

$$(\mathcal{K}_{\text{x}} + \mathcal{K}_{\text{x,inc}})|\Delta x_{\text{lim}}| = \mathcal{K}_{\text{x}}|\Delta x + \Delta x_{\text{inc}}| \tag{4.1}$$

up to the stiffness saturation limit. This rule can be used to both increase and decrease the stiffness in a certain direction. It is important to note that the mentioned operations are carried out in each of the principal Cartesian directions. Let us assume that $\mathcal{K}_{\text{x}} > \mathcal{K}_{\text{mean}}$ meaning that also Δx is saturated, i.e. $|\Delta x| \approx |\Delta x_{\text{lim}}|$; so if Δx_{inc} has an opposite sign than Δx, i.e. if the directional feedback points in the opposite direction that the robot is pushing right now, then $|\Delta x + \Delta x_{\text{inc}}| / |\Delta x_{\text{lim}}|$ is lower than one. Hence also $(\mathcal{K}_{\text{x}} + \mathcal{K}_{\text{x,inc}}) / \mathcal{K}_{\text{x}}$ has to be lower than one, implying that $\mathcal{K}_{\text{x,inc}}$ has to be negative. If Δx_{inc} has the same sign as Δx, the correction to the stiffness model will be positive, i.e., $\mathcal{K}_{\text{x,inc}} > 0$.

This approach not only simplifies the feedback modality but also facilitates the teaching of force tasks with abrupt discontinuities. For example, in the scenario of cable unplugging, a closer attractor with a higher stiffness helps to prevent the “recoil" effect upon object separation. Similarly, if we are pushing a heavy box, the limitation of the attractor distance bounds the robot velocity in case the contact point with the box is lost (see Fig. 4.5). This allows a safer coexistence of the robot in anthropocentric environments.

Finally, in the interactive session, for the purpose of explicitly labeling the desired goal point with zero velocity and high stiffness, a further teleoperation input was employed.

4.3.3 Stabilizing Attractive Field

External forces can lead the robot end-effector in previously unvisited regions of the workspace where the extrapolation of the desired $\Delta \boldsymbol{x}$ and $\mathcal{K}$ can have high uncertainty and lead to dangerous and undesired dynamics of the robot. This problem, known as covariate shift, is common when applying Behavioral Cloning, and some solutions like Disturbances for Augmenting Robot Trajectories (DART) [102] or HG-DAgger [87] investigated the injection of noise in the supervised policy execution in order to lead the robot in unvisited regions and collect a database in a larger portion of the environment. This technique could also be applied in the Interactive Corrections segment (Fig. 4.1b), however, collecting many correction points can be time-consuming and highly data inefficient.

As an alternative, we want to exploit the information of the model variance and its continuous differentiability for modeling how to reject external disturbing forces, i.e., not related to the manipulation operation. Intuitively, we can imagine the variance manifold as a manifold with a furrow that is generated in proximity of the labeled regions of the workspace, as we do when we create the circuit for a marble race on the beach. In the absence of external disturbances, the end-effector would lay in the regions of minimum variance and move inside there. However, the robot should reject forces that are leading its motion to a region of uncertainty, proportionally to the rate of change of the same measure. Equivalently, when the external forces are not disturbing the motion anymore, ideally, the robot should go back to regions where the predictions have higher confidence. It is similar to adding a gravitational term in the variance manifold inducing the end-effector to always “fall" into regions of minimum variance, as a marble would come back on track when disturbed by any collision.

The implementation of this stabilization prior is straightforwardly a force field that is proportional to the gradient of the variance manifold according to:

$$\boldsymbol{f}_{\text{stable}}(\boldsymbol{x}) = -\alpha \nabla(\sigma^2)$$

where the gradient is computed according to Eq. (3.10) and α is an automatically modulated constant according to a maximum allowed force, which ensures that $\boldsymbol{f}_{\text{stable}}$ is never higher than the set threshold. Since the ARD kernel has different lenghtscales in the different Cartesian directions, this turns into a different rejection behavior in different Cartesian directions. For example, if the model gets uncertain faster in a certain Cartesian direction, it will act with a stronger force in that direction because it does not want to go into regions where it would not know how to act. Additionally, the use of this prior also results in another interesting behavior when multiple demonstrations are provided. In the regions of lower overlapping of the demonstrations we can imagine a broader furrow in the variance manifold compared to regions of highly overlapped demonstrations. In the first case, there is a larger track where the “marble" can move before reaching the borders and getting pulled down; on the contrary, in the second case, the narrow furrow forces the robot to stay closer to the overlapping demonstrations. This different behavior of reacting to external disturbances can be interpreted as adaptive disturbance compliance of the robot along the lines of [82] , where higher variance in the demonstration results in higher robot compliance and vice-versa.

4.3.4 Stiffness and Attractor Modulation

Finally, before sending the desired attractor and the stiffness to the robot, we want to make sure to spread the effect of $\boldsymbol{f}_{\text{stable}}$ as stiffness and attractor modulation in order to respect the constraint of having a limited attractor and to obtain the desired force with an increase of stiffness, in each of the Cartesian directions. The desired force is $\boldsymbol{f} = \mathcal{K}\Delta\boldsymbol{x} + \boldsymbol{f}_{\text{stable}}$, so we must find the new stiffness and attractor in each direction such that to satisfy the following equality:

$$\mathcal{K}_{stable}\Delta\boldsymbol{x}_{stable} = \mathcal{K}\Delta\boldsymbol{x} + \boldsymbol{f}_{\text{stable}}.$$

We first set the modulated stiffness as the stiffness predicted by the GP and we compute the new attractor displacement, i.e.,

$$\Delta\boldsymbol{x}_{\text{stable}} = sat(\Delta\boldsymbol{x} + \mathcal{K}^{-1}\boldsymbol{f}_{\text{stable}})$$

where we saturate the attractor magnitude in each principal direction, similarly to Eq. (4.1), to not be larger than $|\Delta x_{lim}|$. Then, we compute the actual desired stiffness magnitude in each principal direction in order to apply the right force considering the stabilization field. For example, in the x direction,

$$\mathcal{K}_{\mathrm{x,stable}} = \frac{|\mathcal{K}_{\mathrm{x}}\Delta x + f_{\mathrm{x,stable}}|}{|\Delta x_{\mathrm{stable}}|}$$

Additionally, when the robot is in a position where the uncertainty approaches the maximum, it is safer to pull the robot stiffness down to zero, rather than the predicted mean value of the GP, according to

$$\mathcal{K} = \mathcal{K}\left(\frac{1-\sigma^2/\sigma_p^2}{1-\beta}\right) \text{ when } \sigma^2/\sigma_p^2 > \beta, \tag{4.2}$$

where σ_{p}^2 is the variance of the prior GP, and β is the relative uncertainty threshold. These two operations are summarized in lines 24 of Alg. 1. Thanks to the property of distance-based kernels of having the prediction to vanish in high-uncertainty regions, and the modulation of the stiffness, the risk of moving in unknown regions of the workspace with possible undesired behaviors is mitigated. Moreover, this behavior results in the robot stopping with high compliance and can be seen as a non-verbal request of teaching, or repositioning into regions closer to the demonstration. Finally, this property also circumvents the issue of variable stiffness instability [97] [50] making the growing oscillations around the nominal trajectory mitigated. This would ensure a safe interaction with the user and the environment.

4.3.5 Learning Nullspace Control Policy

For a redundant robot, it is possible for the end-effector to return to the same task-space position and yet the robot to be in a completely different joint configuration. This would result in unpredictable behaviors with consequent frustration of the human teacher. Unfortunately, this is generally the result obtained when methods based on Cartesian impedance control are used to control the robot's motion. To solve this problem, we also learned a nullspace control policy (always from demonstrations) and had it running during the normal execution of ILoSA. The nullspace action is executed by setting the desired joint angles and joint stiffness as explained in Eq. (A.2).

During the kinesthetic demonstration, the robot's cartesian position and the first 4 joint angles are recorded; then a GP is fitted to map the input cartesian position with the desired joint configuration. Since no prior configuration is set, i.e. the prior joint configuration is the zero vector, to avoid attracting the joint to an undesired configuration, the stiffness modulation of Eq. (4.2) is used with a low threshold ($\beta = 0.2$). This implies that when the end effector is dragged in an unknown Cartesian position, the nullspace stiffness drops quickly to zero. It is worth mentioning, that the minimization of variance will automatically bring the robot close to the region of high certainty, making the Cartesian control and the nullspace control active again. During policy execution, this guarantees the cyclicity of the operations, i.e. the robot coming back to the same Cartesian position with the same joint configuration.

4.4 Real-Robot Validation Experiments

In order to validate the proposed approach, we conduct experiments on four different manipulation tasks, each with its own variations, intended to test the different aspects of ILoSA. The first involves removing a plug from its socket and bringing it to a specified

	Demo Time [s]		**Feedback Time [s]**		**Data Efficiency [%]**		**Goal Error [m]**	
	Single	Multiple	Single	Multiple	Single	Multiple	Single	Multiple
Max	8.00	28.27	3.47	2.13	97.06	97.06	0.016	0.030
Mean	6.73	23.84	1.96	1.61	95.36	96.10	0.009	0.014
Min	5.67	21.40	1.27	1.40	92.86	95.65	0.003	0.008

Table 4.1: Performance in Unplugging.

	Goal Error [m]	
	Without Stabilization Prior	With Stabilization Prior
Max	0.756	0.040
Mean	0.337	0.033
Min	0.073	0.019

Table 4.2: Effect of Stabilization.

goal (Sec. 4.4.1). In this scenario, the effect of the number of demonstrations in broadening the variance furrow is tested. Additionally, the effect of the stabilization prior to rejecting disturbances is analyzed by carrying out the control, in one case with the prior active and in another without it, all while injecting a randomized force disturbance. The second scenario is pushing a box to a goal (Sec. 4.4.2) and observing the handling of contact loss. To test this, an ablation study was carried out. In a further experiment, a periodic perpetual movement scenario in the form of cleaning a whiteboard is analyzed (Sec. 4.4.3). This scenario brings with it the additional challenge that the desired attractor position is located behind the board. Due to the lack of a force sensor between the hand and the tool, this cannot be inferred from kinesthetic demonstrations. The torque sensor in the joint can be used to estimate the external force, see Sec. A.3; however, during kinesthetic demonstrations, the external force applied by the human to move the robot and the reaction from the surface cancels out and cannot be detected from the sensors. The only way to estimate that is to have a sensor between the robot hand and the cleaning tool.

Instead, user corrections are required for the robot to learn to exert the required force on the board for successfully cleaning it. An additional challenge was addressed to ILoSA in this scenario: validate the flexibility of altering a taught behavior to new situations. To showcase this, an obstacle was placed along the original trajectory, limiting the possible height of the motion. The aim was then to provide corrections such that the robot would perform the task while also avoiding the obstacle. Lastly, we evaluated whether the framework could be utilized for more precision-critical tasks (Sec. 4.4.4). For this, a validation experiment is carried out wherein the robot was taught to insert a plug into a socket by non-expert users.

All of the experiments are carried out by the authors five separate times. For the experiments in the second scenario which involved non-expert participants, the participants performed the task twice. The first was a trial round in order to get familiar with the teleoperation device before the second, official trial round.

For the experiments, we utilize the 7 DoF Franka-Emika Panda with an impedance controller and a ROS communication network for the online update at 100 Hz of the attractor and stiffness using the ILoSA framework. A 3Dconnexion SpaceNavigator (see Fig. 4.2) was used for providing teleoperation feedback, whereof one of the two buttons (seen circled

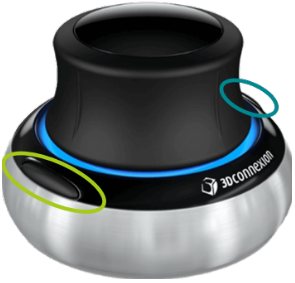


Figure 4.2: SpaceNavigator. The 3D mouse is used to give feedback on the Cartesian attractor and stiffness of the manipulator. The buttons are used to activate/deactivate the human control or to mark the final goal of the policy such as to label it with higher stiffness.

in Fig. 4.2), was used for explicitly marking the desired goal position as noted at the end of Sec. 4.3.2. For all of these experiments excepting the task of inserting a plug, the following parameters were chosen within ILoSA; a mean stiffness of $K_{mean} = 600\,\mathrm{N\,m^{-1}}$ with the stiffness limited to $K_{max} = 2000\,\mathrm{N\,m^{-1}}$, a maximum attractor distance of $0.04\,\mathrm{m}$ along each principal axis, the epistemic uncertainty threshold for adding new points to the database set to $\sigma^2_{Threshold} = 0.2\sigma^2_p$, and the threshold for modulating the stiffness, $\beta = 0.99$. The squared exponential kernel was selected within the GP models. In the case of plugging, the only change to the parameters was the limitation of the stiffness to $K_{max} = 4000\,\mathrm{N\,m^{-1}}$, and the maximum attractor distance to $0.02\,\mathrm{m}$.

For quantifying data efficiency, we compute the ratio between the amount of corrections that result in an increase in size of the existing database and the total amount of provided feedback inputs. The feedback time was computed as the amount of time explicitly spent providing corrective inputs. A data efficiency of 100 % means that we are only updating the old labels and not aggregating new ones.

4.4.1 UNPLUGGING

Two variations of this scenario were performed. In the first, three separate demonstrations were carried out with different heights of the trajectory towards the goal. In the second, a single demonstration was provided. The primary focus of the corrections is placed on the successful unplugging as well as reaching the goal within a tolerance of $3\,\mathrm{cm}$. A standard type F plug was used for which specific 3D-printed gripper jaws were designed to ensure a firm grip throughout the interaction. The interaction commences from the point in which the robot is already gripping the plug. Fig. 4.3 visualizes examples of the resulting attractor fields generated from $\mathcal{K}\Delta\boldsymbol{x}$ for both single and multiple demonstrations. As expected, the highest forces are exerted at the beginning, during the unplugging. Instances of moderate forces can be observed leading towards the trajectories. In particular, for the single demonstration, moderate forces are close to the demonstration itself and are, in fact, directed towards the demonstrated trajectory. For the multiple demonstrations these moderate forces are primarily present outside the region of demonstration. This is attributed to the larger variance furrow, which reduces the effect of the stabilization prior in the demonstrated region, and in turn enables the robot to move more freely within the region when perturbed.

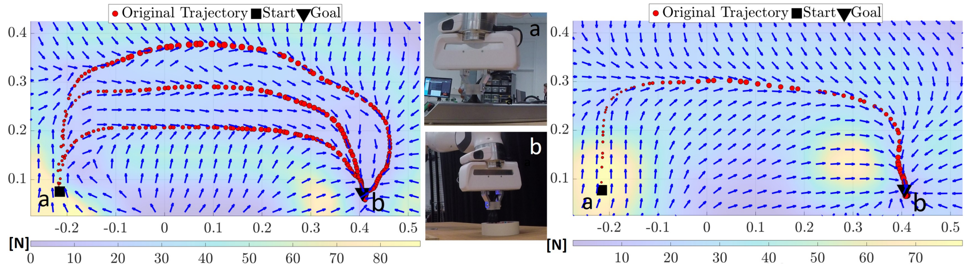


Figure 4.3: Example of attractor fields for unplugging with multiple demonstrations (left) and a single demonstration (right).

The results regarding the precision in reaching the goal indicate that for both variations, the robot was able to successfully complete its task. Slightly larger errors in the case of multiple demonstrations can be seen. This can, however, be attributed to the variations in the final positions provided during the multiple demonstrations. The time spent giving corrections to complete the task successfully was similar between the two experiments with an average time of 1.96 s for the single demonstration and an average time of 1.61 s for the case with multiple demonstrations. For additional details, refer to Table 4.1. For both experiment variations, the majority of feedback inputs did not increase the size of the database, showing in both cases a high data efficiency of more than 95% on average.

In the tests with randomized perturbations, the disturbance was sampled for each of the three axes from a normal distribution $\mathcal{N}(10,5)$ N at 1/3 of ILoSA's update frequency. Here, the benefit of the stabilization prior is clear. When using the stabilization prior, the error from the goal was on average ten times lower than when the prior was not present. When using the prior, despite the perturbations, the robot remained close to the goal, with the highest observed error being 4 cm, indicating high robustness. Furthermore, when the prior was not present, the robot diverged in 3 of the 5 trials and was unable to reach the vicinity of the goal. Additional details are presented in Table 4.2.

4.4.2 Pushing a Box

In real-world scenarios it can easily happen that objects are unwieldy for the robot's gripper, such that the only remaining option for manipulating said objects is pushing them. In such an interaction, it can happen that the object is removed prematurely, resulting in an unexpected contact loss for the robot. ILoSA enables the limitation of the observed velocities in such a case by limiting the maximum attractor distance. In contrast, if the force were to be achieved by increasing the attractor distance while maintaining a low constant stiffness, these velocities could be noticeably greater. To verify this, the task was first learned with ILoSA, wherein both stiffness and attractor distance are variable; afterward, it was learned with a variation of the ILoSA algorithm, wherein the force is altered only through a variable attractor distance which is left unbounded. In both the cases, once the interaction was learned, the task was executed with the box being removed while it was being pushed.

Fig. 4.5 displays the resulting velocities when varying solely the attractor distance (red), as opposed to concurrently varying the attractor distance and stiffness (blue). As can be seen, the peak velocity for the combined variation of both stiffness $\mathcal{K}$ and attractor distance

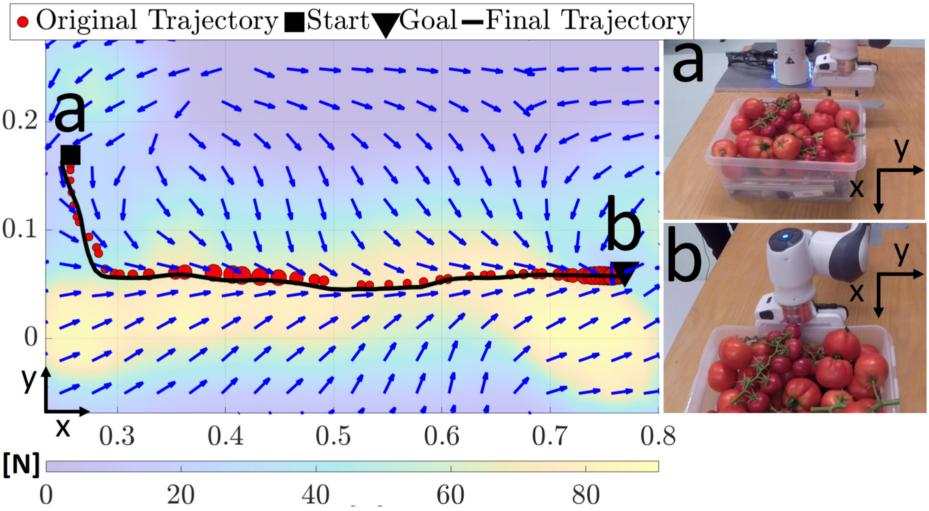


Figure 4.4: Example of an attractor field for the box pushing task.

	Demo [s]	Fdbk [s]	Eff. [%]	Goal Err. [m]
Max	6.80	4.53	98.57	0.016
Mean	5.23	4.16	95.82	0.008
Min	4.47	3.87	90.00	0.001

Table 4.3: Performance in Pushing a Box.

$\Delta \boldsymbol{x}$ is less than half compared to only varying the attractor distance. This in turn allows the limitation of potential impact forces should a person cross the robot's trajectory.

In terms of performance, ILoSA achieves comparable results to those observed in unplugging, both in terms of error from the goal and data efficiency. The overall correction duration as well as the correction duration relative to the one of the demonstration are larger than those observed when unplugging. This is, however, primarily attributed to the fact that the box pushing scenario has a larger portion of the trajectory in which force has to be applied, in turn requiring more user corrections. An example attractor field can be seen in Fig. 4.4. On the matter of data efficiency, it should be noted that both in the case of unplugging and in the case of the box, a data efficiency of 100% was not achieved due to the goal conditioning, which adds the marked goal to the database. For an overview on the task performance, refer to Table 4.3.

4.4.3 Cleaning a Whiteboard

In this task, the robot is taught to ensure a whiteboard remains clean and to sustain the movement until the controller is stopped. For this, it is highly desirable that at the end of each operation cycle, the robot returns to the same joint configuration; this property is known as "cyclicity" of motion [39]. The execution of this task was deemed successful if the desired

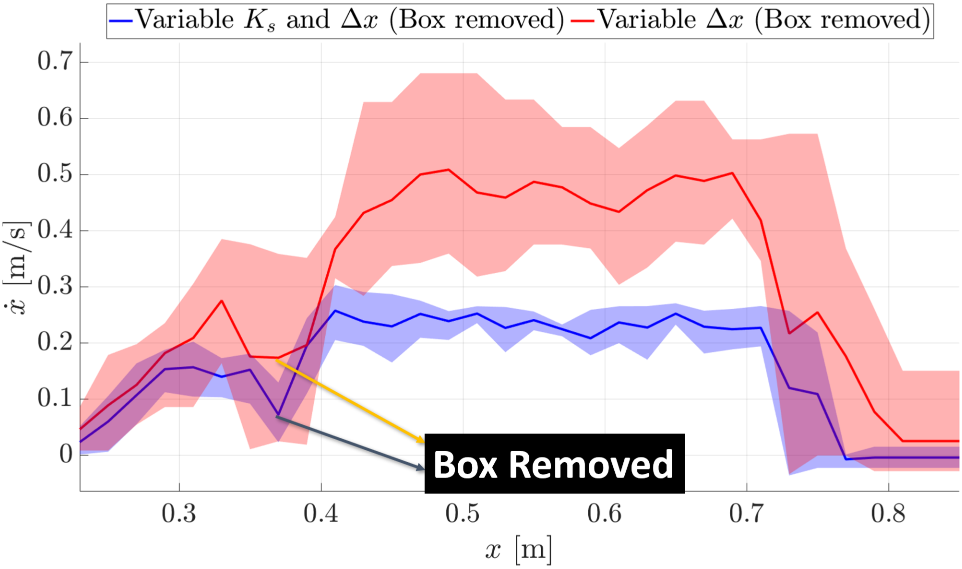


Figure 4.5: Robot velocity w.r.t. current position along trajectory.

	Demo	**Fdbk [s]**		**Data Eff. [%]**		**Consist. [m]**	
	[s]	Orig.	Adap.	Orig.	Adap.	Orig.	Adap.
Max	18.27	6.4	8	100	91.67	0.004	0.004
Mean	16.81	4.81	5.57	98.54	83.14	0.003	0.003
Min	15.20	3.53	3.27	94.51	73.47	0.003	0.003

Table 4.4: Performance in Cleaning a Whiteboard.

area of the board was wiped clean after each loop and the motion continued for at least 5 loops. An example of the resulting attractor field can be seen in Fig. 4.6.

On the quantitative side, not only were the 5 loops executed successfully, but the robot also remained highly consistent in its motion. The consistency was measured as the highest RMSE between each pair of the five loops and it amounted on average to 0.36 cm. Part of the success of a repeatable motion can be credited to the nullspace control that ensures a cyclic joint configuration and successively a consistent Cartesian mass matrix and dynamics. In terms of correction time, only a short period was spent providing inputs, with an average time of 4.81 s. Out of these inputs, the majority resulted in modifications of the database attaining an average data efficiency of 98.54%. Additional details are provided in Table 4.4.

When correcting the original trajectory, adaptations for avoiding the obstacle could be carried out in all five trials. With this, we want to show how, when close to the originally provided samples, the corrections can effectively modify the database to address the desired behavior, and how, when outside the region of certainty, new samples are added, allowing the reshaping of the stabilization field around them. In Fig. 4.7, it is possible to see how around the newly added points the force field would reject the disturbances and stay on the

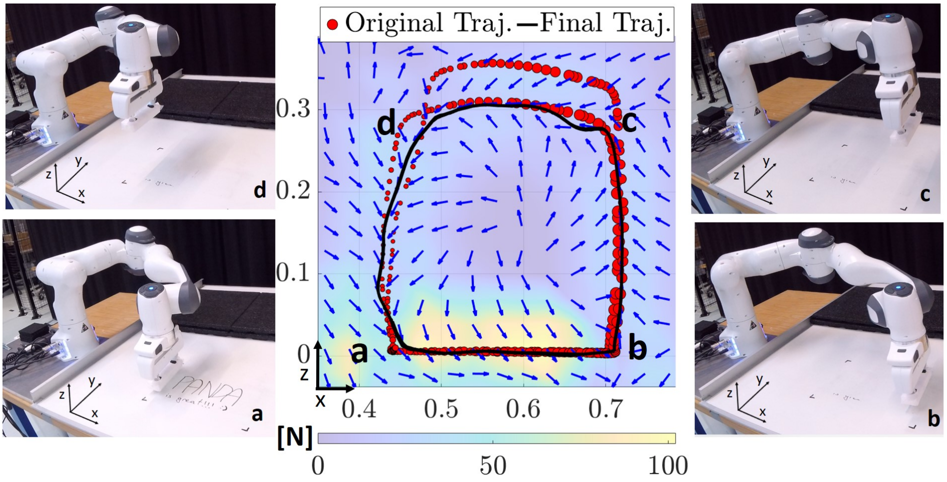


Figure 4.6: Example of an attractor field for wiping a board.

	Fdbk [s]	Total Time [s]	Rounds of Correction
Max	17.49	193.85	6
Mean	4.72	95.84	2.07
Min	1.22	54.46	1

Table 4.5: Performance in Inserting a Plug.

new desired trajectory. Through an additional 5.57 s of corrections on average, and with an average data efficiency of 83.14%, the motion was successfully adapted, resulting in attractor fields similar to the one seen in the figure.

4.4.4 Evaluation with Non-Expert Users: Plugging

After observing that the framework was capable of remaining in close vicinity of the demonstrated trajectory, the task of plugging was taken for its clear constraint on the goal position and the requirement to apply force in order to successfully accomplish the task.

In order to verify that the successful completion of the task was not dependent on executing the training in a specific manner, non-expert participants were asked to carry out the experiment, see Fig. 4.8. A total of fifteen participants aged 23 - 32 took part in an additional engineering validation, rather than a dedicated human factors study. The participants were allowed to perform the task twice. The first time was to give participants the opportunity to get familiar with the teleoperation device for a couple of minutes. The second time was then treated as the official trial, wherein the participants provided a new demonstration and were allowed to provide as many rounds of correction as needed until the robot carried the task out successfully.

All participants were able to complete the task in a fairly short period of time. On average, the total time needed to train the task was about one and a half minutes, which includes both the time needed for providing the kinesthetic demonstration as well as any

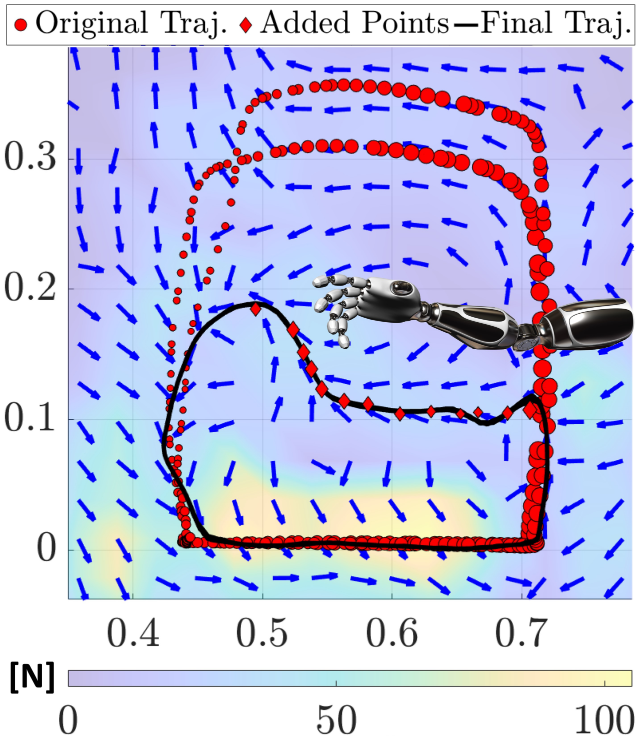


Figure 4.7: Vector-field and the 5 overlapping final trajectories (black line) after user corrections for learning to avoid the obstacle represented by the arm in the picture.

amount of correction rounds needed for the robot to successfully insert the plug. A summary of the performance metrics can be viewed in Table 4.5. In terms of data efficiency, all of the participants completed the task with 100% data efficiency, indicating that the robot never left the region of the demonstration.

Additionally, participants were asked to fill out the NASA TLX and Van der Laan questionnaires after completion of the task. A majority of the participants reported low mental demand and found the teaching method overall helpful and easy to use.

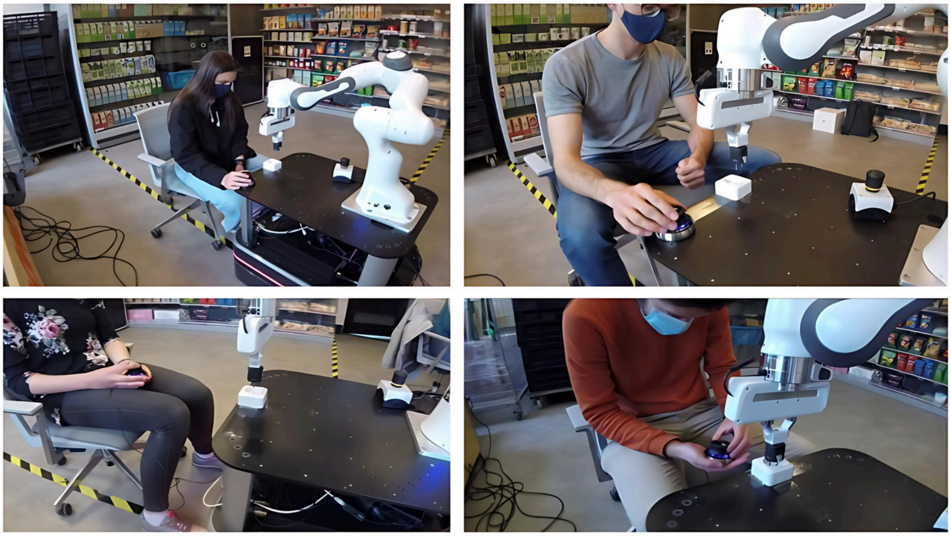


Figure 4.8: Four non-expert users using ILoSA to first demonstrate and then correct the task of inserting a plug.

4.5 Conclusions and Future Work

We have introduced a non-parametric and interactive approach to learning different types of force interaction tasks from humans, while exploiting impedance control to ensure safe interaction. All aspects of the interactions, from the attractor distance and stiffness at the end-effector to the nullspace control, were successfully modeled with the help of GPs. This enables non-experts to create complex robotic interaction skill that would alternatively require a set of expert skills through manual programming.

Making use of the learned model parameters, it was possible to establish two additional safety features. The first is the reduction of the stiffness should the robot be too far from the demonstrated region, eventually bringing it to a halt. The second is a stabilization prior, which helps to steer the robot back to the closest area with low variance, consequently returning it back to the demonstrated region. As a result, this enables the rejection of disturbances.

Moreover, the stabilization prior was able to infer that its effect should be reduced in the areas between demonstrations, allowing the robot more freedom of movement in those areas. However, in the case of desired multimodal behaviors, e.g., for obstacle avoidance, this could be obtained with a constraint on the maximum lengthscale of the used kernel. This is equivalent to the generation of multiple separate variance furrows rather than a single wider one.

These investigations further showed that the ILoSA framework can be implemented for carrying out both goal-oriented and periodic movements. When used in combination with a nullspace control, which enabled cyclicity, a high consistency of the motion was attained. Overall, ILoSA exhibited good reliability in the execution of the examined tasks while learning in a user-friendly and data-efficient manner.

Due to the successful applications in the force tasks in which it was tested, ILoSA will be extended to further challenges in the field of robot manipulation. The learning will not only focus on a single trajectory but on the assembly of movement sequences for more complex tasks, always learning from demonstration. Adaptation of the motion with respect to a particular reference frame in each segment will be investigated using human feedback and the information on the model confidence for solving possible ambiguity in Ch. 8 or adaptation to different surfaces to clean in Ch. 7. In the next chapter, we will extend the framework on learning and correcting the pose and the gripper dynamics using GPs to investigate the task of picking at non-zero-velocity.

5

Interactive Learning to Pick at Non-Zero-Velocity

This chapter investigates how robots learn the intricate task of a continuous Pick ad Place (P&P) from humans based on demonstrations and corrections. Due to the complexity of the task, these demonstrations are often slow and even slightly flawed, particularly at moments when multiple aspects (i.e., end-effector movement, orientation, and gripper width) have to be demonstrated at once. Rather than training a person to give better demonstrations, non-expert users are provided with the ability to interactively modify the dynamics of their initial demonstration through teleoperated corrective feedback. This, in turn, allows them to teach motions outside of their own physical capabilities. In the end, the goal is to obtain a faster but reliable execution of the task. The presented framework, Minimum Uncertainty Dynamical System (MUDS), learns the desired movement dynamics based on the current Cartesian position with GPs, resulting in a reactive, time-invariant policy. Using GPs also allows online interactive corrections and active disturbance rejection through epistemic uncertainty minimization. The experimental evaluation of the framework is carried out on a Franka-Emika Panda. Tests were performed to determine i) the framework's effectiveness in successfully learning how to quickly pick and place an object, ii) the ease of policy correction to environmental changes (i.e., different object sizes and mass), and iii) the framework's usability for non-expert users. A video of the experiments can be found here: `https://youtu.be/XoW6AkK793g`.

This chapter is based on *Mészáros, A.,* ***Franzese, G.****, and Kober, J. (2022). Learning to pick at non-zero velocity from interactive demonstrations. IEEE Robotics and Automation Letters, 7(3), 6052-6059* [116].

5.1 Introduction

More often than not, robots employ a P&P strategy wherein they approach the object, stop and grip it, and only then resume moving. We as humans, on the other hand, tend to pick things in a single fluent and quick motion. Of course, robots should also be able to complete a task fairly quickly, which in the case of P&P introduces a number of challenges, both from a control point of view [166] as well as a learning point of view [18].

Learning from Demonstration (LfD) has become a popular approach for allowing non-expert users to teach robots and thus more easily integrate them into the working and daily environment [137]. Yet these provided demonstrations are sub-optimal compared to what the robot might be able to achieve, e.g., demos having slower dynamics. Concurrently, it is important to consider that often, the execution of a task cannot simply be sped up uniformly. For example, when learning a P&P movement, retaining a high velocity when approaching the object can generate high impact forces which can cause the object to bounce away or topple over, potentially damaging the item in question as well as making it impossible to pick on time. We as people are able to identify such constraints and adapt accordingly, and can transfer this knowledge to the robot through demonstrations.

This work studies the feasibility of robot picking only using time-independent policies learned from human demonstrations and corrections. The previous chapter already revealed the effective application of minimum uncertainty GPs for learning variable impedance control in force application tasks like cleaning, plugging, and pushing. In none of the previous cases, however, were the dynamics of the End-Effector (EE) orientation or gripper learned nor were there critical contact dynamics involved. Teaching more degrees of freedom while asking for fast performance makes the task of non-zero-velocity picking a challenging benchmark for studying the potential of learning from non-expert human teachers.

The main contributions of this chapter over the previous are:

1. Proposing a framework for interactively altering the speed and shape of robot motion dynamics in a decoupled manner through teleoperated correction.

2. A novel minimum uncertainty inference for learning the desired non-linear constraints of EE orientation and gripper width w.r.t. the EE position dynamics, while avoiding dangerous extrapolations.

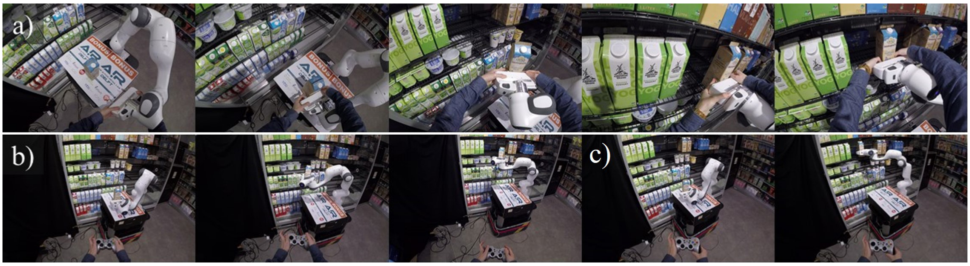


Figure 5.1: Learning flow for teaching a robot how to reshelve an item.; a) starting with a single demonstration, followed by b) multiple rounds of correction, after which, c) the robot can carry out the task autonomously.

3. Showing the benefit of uncertainty minimization for enabling local motion consistency when dealing with critical precision tasks like fast picking, while being compliant in the interaction.

4. Extending the framework for generalizing to different object positions thanks to the parametrization w.r.t. moving reference frames.

Fig. 5.1 summarizes the three phases of learning in the teaching of a reshelving operation: the initialization of the policy with kinesthetic demonstration, the shaping of the dynamics with teleoperated corrections, and the final evaluation of the autonomous task execution.

5.2 Background and Related Work

When executing high-speed manipulation tasks that involve establishing contact with an object, it is important to consider the behavior around the moment of impact. A reoccurring approach observed in existing works consists of adapting the relative velocity in order to mitigate the effects of the impact [169]. Another strategy, which has been employed to absorb impacts, particularly in catching tasks [89], involves utilizing a follow-through behavior that continues to track the predicted path of an object even after interception [143]. Alternatively, one can incorporate compliant behavior into a provided attractor using impedance control [21]. While it is unable to mitigate the initial impact force irrespective of the set stiffness since the main contribution to this force is the velocity of the impacting objects, it is beneficial for absorbing the post-impact forces [67].

We can conclude that matching the velocity of an object likely achieves the best reduction of impact force, however, such an approach may not be optimal when considering the total time of the trajectory execution. This is especially true for static objects, wherein matching velocities would effectively bring the robot to a standstill prior to the picking action. A better approach, therefore, is to interactively learn the feasible non-zero contact velocity while ensuring moderate impact forces.

Being able to adapt/correct the learned velocity with ease plays a key role in speeding up the overall execution of the demonstrated trajectory while also considering that the movement dynamics may require different degrees of adaptation at different points of the trajectory; for example, slowing down prior to the moment of interception. Different works explore speed adaptation during trajectory execution using different function approximators. One approach involves altering the phase rate of probabilistic movement primitives (ProMPs) [94], whereas others propose a modified version of Dynamical Movement Primitives (DMPs) in which the speed is altered through an additional phase-dependent temporal scaling factor [120], or where the temporal scaling factor is changed through corrections and subsequently translated to changes in the learned dynamic movement [86]. The mentioned works modulate the velocity either using optimization approaches or defined functions or in the case of [86] where human corrections are used, the corrections are provided in a coupled manner for both the trajectory shape and speed. Our approach instead focuses on combining imitation learning and human interactive feedback [38] to provide corrections to speed and shape in a decoupled manner through teleoperation.

An alternative to phase-dependent methods, like DMPs, can be obtained as the formulation of the motion as a reactive controller according to

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) \tag{5.1}$$

where $\boldsymbol{x}$ is the robot state and $\dot{\boldsymbol{x}}$ identifies the transition of the robot state. GPs have been used for shaping a motion from human demonstrations through the local modification of a stable field [98]. However, none of the other works on learning state-dependent dynamical systems take into account the information of the uncertainty to increase motion consistency, and reduce covariate shift. Furthermore, in the context of interactive learning, we introduced the idea of decoupling the corrections of shape and velocity and investigated how this can be beneficial for allowing non-expert users to teach challenging tasks.

5.3 METHODOLOGY

The goal of this framework is to enable a user to teach the robot the desired motion through demonstration and teleoperated correction, see Alg. 2. The robot learns the desired minimum uncertainty dynamical system on the end-effector and the dynamics of the gripper, orientation and width as a function of the current robot position formalized in Sec. 5.3.1. The main aim is to show that it is possible to learn a policy and later correct the velocity so as to achieve and surpass the performance of a skilled demonstrator. All of these aspects are modeled with GPs, allowing interactive corrections of the dynamics and actions online, see Sec. 5.3.2.

5.3.1 LEARNING A MINIMUM UNCERTAINTY DYNAMICAL SYSTEM

A non-linear dynamical system can be described by Eq. (5.1). This type of formulation would fit perfectly in a velocity controller, however, due to the necessity of dealing with impacts — for which an impedance controller is more suitable [67] — we can rewrite the motion dynamics into its integral form, i.e. we are controlling the desired next point of the motion and not the current desired velocity, based on

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \int_t^{t+\Delta t} f(x(\tau))d\tau$$

where $\boldsymbol{x}_{t+\Delta t}$ is the desired attractor position. Since $\dot{\boldsymbol{x}}$ is a function of the current position $\boldsymbol{x}$, the attractor distance $\Delta\boldsymbol{x}$ is going to be a function of the robot position $\boldsymbol{x}_t$, i.e.,

$$\Delta\boldsymbol{x}_t = g(\boldsymbol{x}_t) = \int_t^{t+\Delta t} f(x(\tau))d\tau \ := \boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t.$$

The dynamical system can be seen as an external (and slower) control loop where the attractor position is updated as a function of the robot position while the inner (and faster) impedance control loop simulates the dynamics of a critically damped second-order dynamical system towards the chosen attractor. As an analogy to humans, the slower loop can be seen as the intention update when generating a motion according to the current perceived arm position while the impedance control represents the compliance of the muscles and the joints in the interaction with the environment.

The desired $\Delta\boldsymbol{x}$ is fitted with a multi-output GP that shares the kernel parameters, and using the data of a kinesthetic demonstration and user-provided corrections. The GP models

Algorithm 2 Teaching Framework for Interactive Learning to Pick at Non-Zero-Velocity

```
**a) Kinesthetic Demonstration(s)**
**while** Trajectory Recording **do**
    Receive($\boldsymbol{x}_t$, $\boldsymbol{sin\theta}_t, \boldsymbol{cos\theta}_t$, $w_t$)
    $\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t$
    $\gamma_t = 1$
**end while**
$\boldsymbol{\xi} = (\boldsymbol{x}_{t_0}, \boldsymbol{x}_{t_0+\Delta t}, \ldots, \boldsymbol{x}_{t_f})$
$\Delta\boldsymbol{x}^{\text{demo}} = (\Delta\boldsymbol{x}_{t_0}, \Delta\boldsymbol{x}_{t_0+\Delta t}, \ldots, \Delta\boldsymbol{x}_{t_f})$
$\boldsymbol{sin\theta}^{\text{demo}} = (\boldsymbol{sin\theta}_{t_0}, \boldsymbol{sin\theta}_{t_0+\Delta t}, \ldots, \boldsymbol{sin\theta}_{t_f})$
$\boldsymbol{cos\theta}^{\text{demo}} = (\boldsymbol{cos\theta}_{t_0}, \boldsymbol{cos\theta}_{t_0+\Delta t}, \ldots, \boldsymbol{cos\theta}_{t_f})$
$\boldsymbol{w}^{\text{demo}} = (w_{t_0}, w_{t_0+\Delta t}, \ldots, w_{t_f})$
$\boldsymbol{\gamma}^{\text{demo}} = (\gamma_{t_0}, \gamma_{t_0+\Delta t}, \ldots, \gamma_{t_f})$
Train GPs
**b) Interactive Corrections**
**while** Control Active **do**
    Receive($\boldsymbol{x}$)
    **if** Received Human feedback $\Delta\boldsymbol{x}^c$, $\gamma^c$, $w^c$ **then**
        Correct($\Delta\boldsymbol{x}^c \rightarrow \Delta\boldsymbol{x}^{\text{demo}}$, $\gamma^c \rightarrow \boldsymbol{\gamma}^{\text{demo}}$, $w^c \rightarrow \boldsymbol{w}^{\text{demo}}$)
    **end if**
    $[\Delta\boldsymbol{x}, \sigma^2] = \mathcal{GP}_{\Delta\boldsymbol{x}}(\boldsymbol{x})$
    $\gamma = \mathcal{GP}_{\gamma}(\boldsymbol{x})$
    $w_{\text{des}} = \mathcal{GP}_{w}^{MU}(\boldsymbol{x})$
    $[\boldsymbol{sin\theta}, \boldsymbol{cos\theta}] = \mathcal{GP}_{\theta}^{MU}(\boldsymbol{x})$
    $\boldsymbol{\theta}_{\text{des}} = \text{arctan2}(\boldsymbol{sin\theta}, \boldsymbol{cos\theta})$
    $\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + \gamma\Delta\boldsymbol{x} - \alpha\nabla_{\boldsymbol{x}}(\sigma^2(\boldsymbol{x}))$.
    Send($\boldsymbol{x}_{\text{des}}, \boldsymbol{\theta}_{\text{des}}, w_{\text{des}}$)
**end while**
```

the desired $\boldsymbol{\Delta x}$ as a function of the current Cartesian position $\boldsymbol{x}$

$$g(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$$

and the prediction on the desired attractor distance is Gaussian distributed, i.e.,

$$\boldsymbol{\Delta x} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\boldsymbol{I}),$$

where the vector $\sigma^2(\boldsymbol{x})\boldsymbol{I}$ underlines that the uncertainty in the three principal directions of movement, i.e. $\Delta x, \Delta y, \Delta z$, is the same and that no cross-correlation is modeled, i.e. the three outputs are independent. This is possible by setting the kernel hyperparameters to be shared among the three independent models, see Sec. 3.5. The definition of the mean function $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$ are the same of Eq. 3.2 and 3.3. The bold in $\boldsymbol{\mu}$ is just to underline that the mean model is a vector of dimension three.

Finally, something to consider when learning a dynamical system in a reactive formulation is that the next robot position is a function of the learned desired transition but also the external disturbances. This may lead the robot in a position where its policy is not confident anymore, i.e., high epistemic uncertainty. Depending on where this occurs, the robot may not be able to successfully pick up the object or bring it to its goal and execute its motion. When we, as humans, execute a motion we try to remain in regions where we are confident about what we have learned up to that point. To encode this behavior also in the robot, the dynamical system was superposed with another dynamical system that brings the robot towards regions of low uncertainty. From a control point of view, this results in adding another attractor field that is proportional to the gradient of the variance manifold [55].

This repulsive field can be seen as a *behavioral* stiffness: considering a variance manifold as a potential energy, similar to elastic energy, the robot is always acting towards the minimization of this quantity; similarly, the lower level control, "the muscles", is trying to converge to the attractor in order to minimize its *physical* tension. Thus, the MUDS can be summarized as the position, orientation, and gripper control of the end-effector. The desired attractor position is modulated according to,

$$\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + \boldsymbol{\Delta x} - \alpha \nabla_{\boldsymbol{x}}(\sigma^2(\boldsymbol{x})).$$

However, when learning a complex task like a fluent P&P, the dynamics of the end-effector position have to be augmented with the dynamics of the hand orientation $\boldsymbol{\theta}$, and gripper width $\boldsymbol{w}$. Because in a trajectory the dynamics of the orientation and gripper are coupled with the dynamics of the end-effector, we decided to learn the controlled action as a function of the robot's position with another GP. However, if the predictions are done based on the current position, when outside of the region of certainty, the robot would output the prior mean e.g., zero radians for the orientation along all three axes and maximum gripper width, which could lead to an undesirable generalization, e.g., tilting or dropping objects. In order to solve this problem, we propose a *minimum uncertainty inference*, obtained by projecting $\boldsymbol{x}$ in the highest correlated sample of the database according to

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}_i \, \in \, \xi}{\operatorname{argmax}} (\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}_i))$$

where $\boldsymbol{K}$ is built with the kernel function with the optimized hyper-parameters. This minimum uncertainty inference can be interpreted as a projection of the robot's current state on the highest correlated state (according to the kernel function) collected during the demonstration(s). The aim is to explicitly avoid extrapolating outside the original demonstrated data while still using the property of a smooth regressor of the GP. This behavior also matches the philosophy of actively taking actions that would always minimize the uncertainty of the current robot state. When the evaluation of the GP is performed with this *minimum uncertainty* rule, we denote them with the superscript *MU*.

In order to fit the desired end-effector orientation as a function of the current Cartesian position with a regressor, it is necessary to have a smooth and *continuous* representation of the Euler angles $\boldsymbol{\theta}$, which otherwise exhibit a discontinuity when crossing the $[\pi, -\pi]$ boundary. Consequently, $\boldsymbol{\theta} = h(\boldsymbol{x})$ would not satisfy the continuity requirements. To this end, we fit both $\boldsymbol{sin\theta} := \sin(\boldsymbol{\theta})$ and $\boldsymbol{cos\theta} := \cos(\boldsymbol{\theta})$ transformations of the Euler angles and convert them back after the MU inference during robot control (l. 23 of Alg. 2). This also implies that we need to fit six GPs for modeling the orientation rather than three.

5.3.2 Interactive Policy Correction with Human-in-the-Loop

After learning from kinesthetic demonstrations the desired transition $\Delta\boldsymbol{x}$, Euler angles $\boldsymbol{\theta}$ and the gripper width w in the different points of the recorded trajectory, we still need to allow the user to correct the policy during the robot execution.

Our goal is to obtain a fast continuous picking operation. With increasing velocities, kinesthetic interactions with a robot manipulator can become unsafe, and tuning both the attractor and gripper locally becomes very challenging. Furthermore, it also gives rise to ambiguity on the interpretation of the interaction forces as intended corrections or undesired disturbances [86]. For this reason, we opted for teleoperated corrections on the desired movement, local velocity, and gripper width. Thus, due to the necessity of modifying the magnitude of the attractor distance proportionally in *all directions* (when higher/lower velocities are requested), a *scaling* factor is learned as a function of the position, resulting in a desired attractor

$$\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + \gamma\Delta\boldsymbol{x} - \alpha\nabla_{\boldsymbol{x}}(\sigma^2(\boldsymbol{x}))$$

where γ is the attractor scaling factor and is also model as a GP, i.e., $\gamma = \mathcal{GP}_{\gamma}(\boldsymbol{x})$. With this formulation, corrections can be allocated to the 3 different components of the vector $\Delta\boldsymbol{x}$ or to the total magnitude of the vector itself, i.e. γ. The complete control loop with human-in-the-loop corrections can be seen in Fig. 5.2. Overall, corrections are provided to the output values $\boldsymbol{y}_{demo}$ of the different GPs for the attractor distance $\Delta\boldsymbol{x}$, scaling factor γ and the width of the gripper prongs w, all of which are initialized with the kinesthetic demonstration. With the evaluation of the kernel, the corrective input can be smoothly spread to surrounding data points in accordance with their correlation.

The update rule was chosen to be the conditioned update rule of Eq. (3.8) without performing any new aggregation of data. Moreover, the likelihood noise of the teleoperated correction is considered to be zero, i.e. $\sigma_{n,corr} = 0$. Since the multi-output GPs is composed of independent GP, the model update is in each direction independently.

It has previously been shown in Sec. 3.2.2, the spreading the corrections on the database is more user-friendly, as well as time and data-efficient than a simpler data aggregation, since otherwise, the GP model would essentially average between the different outputs for a given input, leading to slow learning. Additionally, this constraint of spreading the corrections only on existing points of the database avoids modifying the shape of the variance manifold, keeping the motion always close to the kinesthetic demonstration, while still shaping the motion dynamics, encoded in $\gamma\Delta x$.

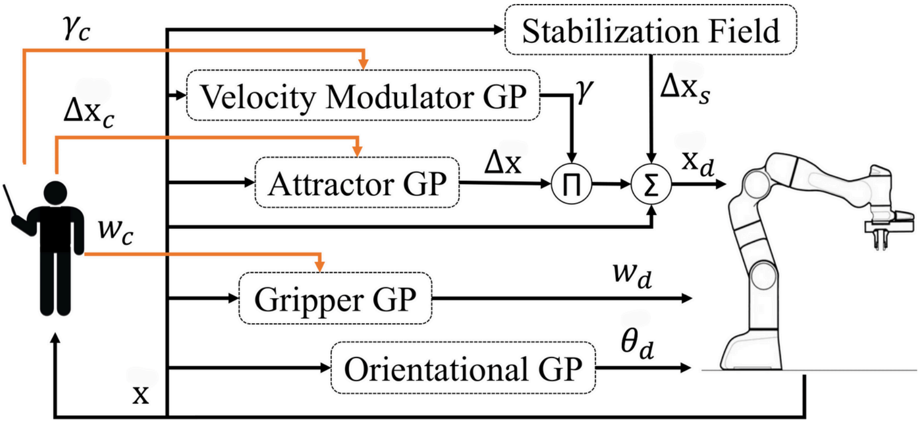


Figure 5.2: A schematic representation of the human-in-the-loop giving corrections to the learned policy. The human observes the current robot motion and gives corrections with a joystick.

5.3.3 Switching Dynamical Systems

The act of grasping an object and taking it to a desired location can be formulated in two parts. The first part is approaching the object and grasping it, and the second, taking the object to its goal. Even if the motion is fluent, these are two subtasks of the overall task that need to be completed. Thus, our attention tends to first be directed towards the object after which our attention shifts to its desired goal location. Following this reasoning, when the demonstration is initially provided, the trajectory is observed w.r.t. the goal and w.r.t. the object location. These observations are used to train two sets of GP models - one for the goal frame and one for the object frame. When seen in the global frame the initial provided observations are aligned as in Fig. 5.3a. If either the goal or the object is moved, the known regions of the two models will no longer be aligned as illustrated in Fig. 5.3b. In order to successfully reach the goal, it is necessary to move towards the known region leading to the goal. The switching of the frames is performed in accordance to the heuristic that once the manipulated object has been grasped, the desire is to bring it to its new location. Furthermore, in order to ensure a smooth switch between reference frames, a short transitioning phase is initiated (Fig. 5.3c).

Firstly, when the model depicting the behavior w.r.t. the goal is selected the uncertainty is minimized with respect to that model's variance, which as a result automatically leads the robot towards the trajectory leading to the goal despite previously not having received any demonstrations in between the known regions. Secondly, in order to avoid abrupt changes in the predictions while in the region between two trajectories, the uncertainty with respect to the currently selected model is utilized to modulate the predictions. The modulation is merely a weighted average between the predictions of the previously selected model and the predictions of the currently selected model. Mathematically this transition function can be

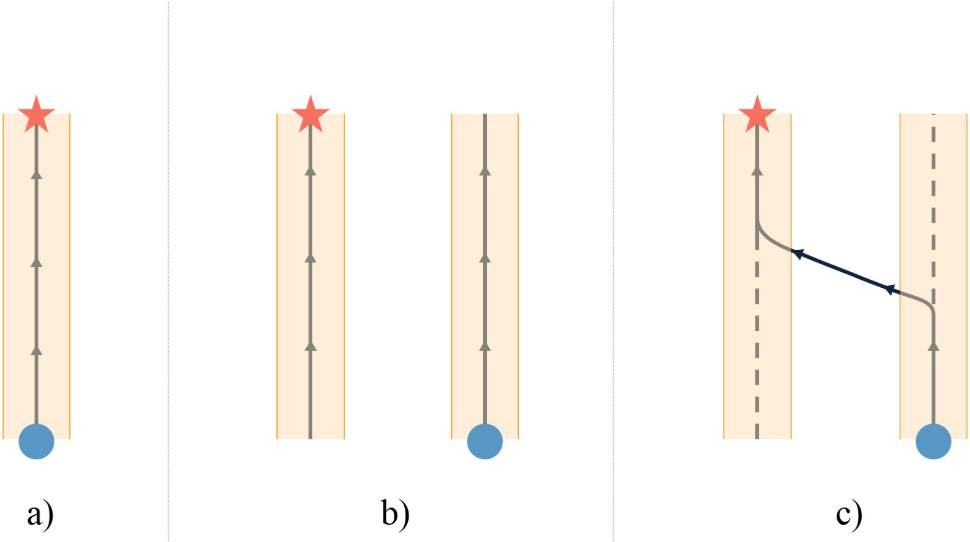


Figure 5.3: a) Initially learned trajectory going from the object (circle) to the goal (star) with the region of certainty depicted by the shaded area. b) Learned trajectories with respect to the two reference frames once they've been displaced. c) Transition enabled through the minimization of the epistemic uncertainty w.r.t. the model leading towards the goal.

written as

$$\boldsymbol{y} = \left(1 - \frac{\sigma_a^2}{\sigma_{\mathrm{p}}^2}\right)\boldsymbol{y}_a + \frac{\sigma_a^2}{\sigma_{\mathrm{p}}^2}\boldsymbol{y}_b$$

where σ_{p}^2 is the variance of the prior of the multi-output GP with the defined kernel, $\boldsymbol{y}_a$ is the prediction of the currently selected model and $\boldsymbol{y}_b$ is the prediction of the previously selected model. This modulation is used until $\sigma_a^2/\sigma_{\mathrm{p}}^2$ falls below a certain threshold, indicating that the robot is once more within the known region. The use of variance minimization ensures that the uncertainty will decrease over time, ensuring the transition from one frame to another.

Only when transitioning between frames are predictions w.r.t. both frames necessary. Otherwise, the predictions are carried out based on the current frame. Furthermore, when corrections are provided, these corrections are applied to the datasets with respect to the different frames, not only the current one. This ensures that knowledge gained in one frame is transferred to the other frame in accordance with the correlation, reducing unpredictable behavior such as abrupt changes in the accelerations during frame switching.

5.4 Validation Experiments

Experiments were carried out to evaluate the effectiveness, usability, and robustness of the method. In Sec. 5.4.1, the framework's base functionality of taking slow demonstrations and allowing the correction of the dynamics through corrective feedback is tested, along with an ablation study to verify the utility of uncertainty minimization. In Sec. 5.4.2, a baseline comparison to a method that also addresses the problem of interactive velocity modulation

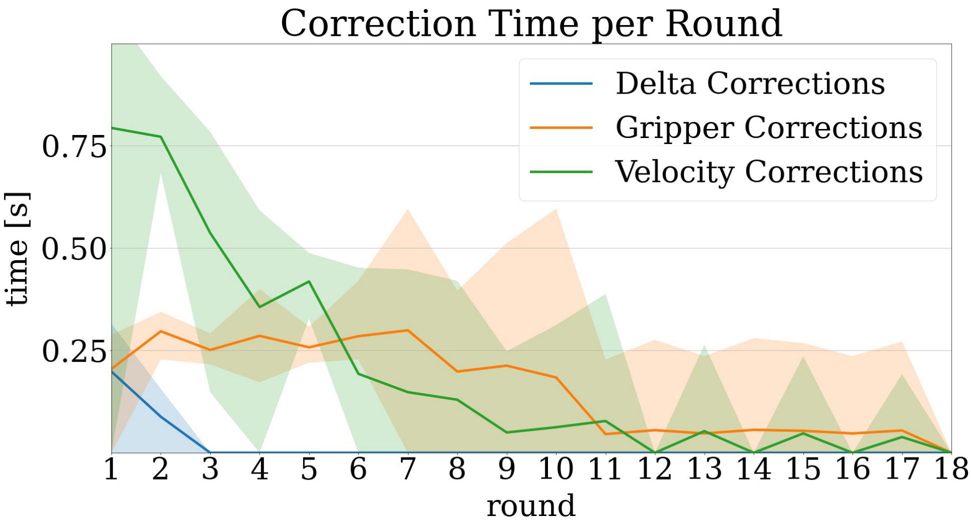


Figure 5.4: Range of correction times per round for each aspect depicted by the shaded areas, with the average times depicted by the solid lines. Statistics made over 5 repetitions.

is performed. Sec. 5.4.3 analyzes how well a learned policy can accommodate changes in object properties such as size and weight. In Sec. 5.4.4, a straightforward generalization w.r.t. different object locations is briefly analyzed. Lastly, in Sec. 5.4.5 a user validation study was carried out with non-experts to establish the usability of the proposed method.

We used the 7 DoF Franka-Emika Panda with an impedance controller and a ROS communication network for the online attractor update with a frequency of 100 Hz. Furthermore, in order to avoid overloading the GP with superfluous data, the recording of the trajectory is carried out at 10 Hz considering that whatever the human is showing at higher frequency is noise that would anyway be filtered out by the GP fitting and the impedance policy.

A wireless Logitech F710 Gamepad was used for teleoperated corrections. The Gamepad was chosen due to the number of required inputs, it being an established ergonomic input device in the gaming industry, as well as ensuring that users remain at a safe distance from the robot at all times considering the high-speed motion dynamics. Due to the limited number of continuous inputs, both the gripper and scaling factor corrections are provided through discrete increments. The attractor corrections are provided through the continuous inputs of the two thumbsticks, with the movement in the x-y-plane regulated by the left thumbstick and the height regulated by the right thumbstick. As an added safety feature, one of the triggers was utilized as a *safety button* which, when released, ends the execution of the algorithm, halting the robot. Lastly, users can comfortably start the execution from any point along the trajectory as well as bring the robot to the start of the trajectory. As a final remark, it is worth underlining that the capability of correcting the orientation after the demonstration was not enabled due to the limitations of the teleoperation interface, not due to any limitations surrounding the algorithm itself, and is thus left to future work.

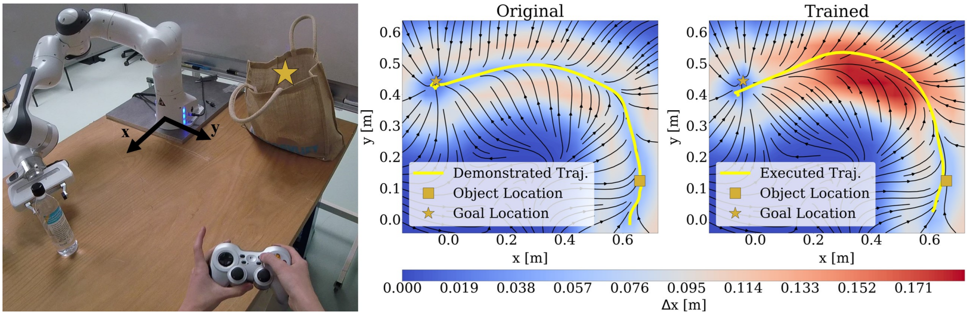


Figure 5.5: Use case of robot assistance in grocery packing. In the attractor vector field, the arrows denote the direction of the attractor and the color gradient denotes the magnitude of the attractor. The vector field based on the original demonstration, with the demonstrated trajectory is compared with the one after training, with the executed trajectory.

5.4.1 Interactive Fluent Pick & Place with MUDS

5

For this experiment, a *single demonstration* was provided wherein the end-effector orientation, gripper width, and attractor distance are obtained and used for initializing the respective GP models. The goal of the task is to i) reduce the execution time by 4 times w.r.t. the demonstration time of the motion with kinesthetic teaching, and ii) have an execution time of 3 s or less. We repeated the experiment 5 times.

Within less than 3 min it was possible to fully train the robot to pick & place the object with the desired performance, four out of five times. Only a fraction of that time was needed for the demonstration (avg. 11 s) and explicit feedback from the human (avg. 6.8 s). This points towards primarily needing fine-tuning corrections from the human, which is further supported by the time spent giving corrections for each of the three correctable aspects (see Fig. 5.4).

It is worth noting that a correction round refers to an execution of a trajectory with optional user corrections, which can be stopped at any point of the execution and not just at the goal. The time spent correcting the attractor was minimal, as it was only required around the moment when the object was reached. This is because the human tends to stop at the object during the demonstration to avoid knocking it over and to deal with the closing of the gripper. To avoid the motion to stop, minor corrections to the attractor were provided to ensure it follows the desired continuous picking motion. Afterward, only corrections for the gripper and scaling factor are provided. Whenever corrections to the scaling factor were provided, resulting in higher velocity, corrections to the gripper had to be provided as well to offset the communication delay of the gripper. Due to the unreliability of the gripper, despite corrections to the timing, the gripper still sometimes closes at the incorrect moment. Nevertheless, after corrections, an average success rate of 82% out of 10 autonomous executions of 5 different trained policies (41 successes over 50 executions in total) could still be achieved. For the complete performance details, please refer to Tab. 5.1.

To verify the existence of gripper unreliability we measured the delay between sending the command for closing the gripper and the actual moment of closing. Measurements were gathered from 20 rollouts. While the average delay was 0.93 s, it ranged from 0.56 s to 1.54 s. Considering this stochasticity, the best strategy is to push the object at non-zero velocity for

	Demo [s]	**Fdbk** [s]	**Total Time** [s]	**Rounds**	**Success Rate** [%]
Max	11.70	10.324	165.44	17.0	100
Mean	10.94	6.796	97.47	10.4	82
Min	10.10	4.560	66.61	6.0	50

Table 5.1: Method Performance (5 demos, 50 executions).

a long enough time so that it encompasses the possible moments at which the gripper might close.

One of the main concerns when increasing the velocity along a trajectory is diverging from said trajectory, particularly in curves. While the shape of the trajectory did change slightly, divergence from the trajectory could be avoided thanks to the uncertainty minimization even when the attractor magnitude was noticeably increased compared to the original demonstration. This can be observed within the attractor vector fields in Fig. 5.5. This is an important feature of the proposed method, opening an alternative to many methods that do not deal with covariate shift when they try to generalize. The goal was to show that even if the dynamics of the trajectory are modified, the obtained trajectory does not change much, resembling the original demonstration.

To further evaluate the benefit of uncertainty minimization in the training as well as in the final execution, we performed an ablation study. The desired policy was trained once with the uncertainty minimization active (w/ UM) and once without it (w/o UM). It was observed that the uncertainty minimization made the training easier since it kept the robot close to the demonstrated trajectory. This translated to a shorter training time of 70 s w/ UM, whereas w/o UM 218 s were needed. We then performed two tests for observing the effect on the execution; one with a perturbation to the robot's initial position and one without such perturbation. The policies were rolled out 20 times each. The effect of the uncertainty minimization was observed in the success rates of the P&P as well as the average distance error (ADE) of the executed trajectory w.r.t. the demonstrated trajectory. Without perturbations, the policy w/ UM achieved a higher success rate of 95% and lower ADE of 0.023 m whereas the policy w/o UM only achieved a success rate of 45% and an ADE of 0.051 m. Similar results are observed when the perturbation is added, where the success rate of the policy w/ UM was 100% and the ADE was 0.034 m whereas the policy w/o UM only achieved 50% and had an increased ADE of 0.090 m.

For an evaluation of its benefit for reaching a goal while rejecting disturbances, the reader must refer to Ch. 4.

5.4.2 BASELINE COMPARISON

We compare to a state-of-the-art approach in interactive dynamics modulation presented in [86]. The base method was replicated based on the details given in the paper with the only major change being that we do not learn the orientation with the DMPs. Since the focus of this baseline comparison was on the modulation of the translational dynamics, the gripper and orientation were controlled with the GPs, conditioned on the robot's current position for all the tests. Corrections were given with the joystick in both cases out of safety concerns when the robot is moving at high velocity.

	Rigid	**Rigid**	**Flexible**	**Small &**
	(250 g)	(900 g)	(100 g)	**def.** (250 g)
	source	**new** | **adp**	**new** | **adp**	**new** | **adp**
Correction Time [s]	51	46 | 0	59 | 38	73 | 24
Rounds	5	5 | 0	7 | 4	8 | 4
Success [%]	88	96 | 98	98 | 100	98 | 96

Table 5.2: Performance in Interactive Adaptation to new object properties.

We initialized the DMP and a version of the proposed algorithm using the scaling factor (*V1*) and without using it (*V2*) with a single demonstration of picking the object given along the y-axis. Then, the object was displaced 7 cm to the side (x-direction) to compare the ability of both algorithms to reshape and speed up the motion.

What could be noticed with the DMP-based approach is that when a correction in x-direction was given the robot would virtually stop and only occasionally move forward. The cause of this was determined to be the dot-product of the position error with the predicted velocity $\widetilde{\boldsymbol{p}}^{\top}\dot{\boldsymbol{p}}_d$. This is used for changing the temporal scaling factor τ of the DMPs, such that when the error is in the direction of the velocity the evolution of the DMP is sped up whereas in the opposite case, the evolution of the DMP is slowed down. In our case, although the predicted velocity along x was very small, it was occasionally negative which could account for the undesired slowing down of the motion. Only in the moments when the velocity became positive along this axis did the robot move forward. As for speeding up, this was later possible along the y-axis, however, the generated acceleration was rather high even when a small correction was given. This is very likely due to the fast convergence of τ. The total training time for a successful picking policy was 197 s. The final achieved execution time was 8.43 s which was 1.17 times faster than the original.

With MUDS the correction in x-direction did not affect the motion in y-direction. Through the correction of the attractor distance along each of the axes, the shape of the trajectory along each of the axes could be easily altered. When using the scaling factor γ (*V1*), the speed along each of the axes of motion increases proportionally. Alternatively, if one chooses to not use γ and directly affect the velocity by changing the attractor distance along an axis (*V2*), one can ensure that the corrections do not affect the remaining axes. Depending on whether the velocity increase should be proportional in all directions (e.g., speeding up a diagonal motion in x-y-direction) or only along a single axis, the two approaches of altering the velocity help account for both possibilities. With *V1* 48 s were needed to train a successful picking policy whereas 46 s were needed for *V2*. The final execution times were 2.67 s for *V1* and 2.24 s for *V2* which translated to an increase of speed by 3.71 and 4.41 times respectively.

5.4.3 Interactive Adaptation to New Object Properties

It can be that we want to pick up a different object after having learned a desired P&P behavior. Even small changes in object properties can result in failure when using the same policy. Rather than demonstrating and retraining the strategy for every new object, or relying on hard-coded rules to adapt to these changes, corrections can be used to adapt the

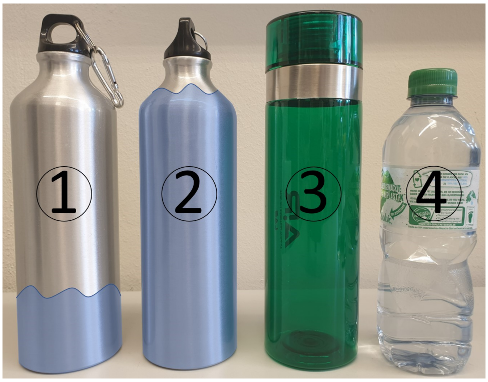


Figure 5.6: L-R: rigid (250 g), rigid (900 g), flexible (100 g), small & deformable (250 g).

learned policy. A selection of four different objects was taken (seen in Fig. 5.6) to make a comparison of training from a new demonstration (new) and training a policy by adapting an existing policy (adp), as reported in Table 5.2.

For the latter case, the initial policy was trained on a rigid water bottle with a weight of 250 g (① in Fig. 5.6), our 'source' object. Once a satisfactory policy was achieved, the training object was swapped out for another object. The policy was then executed and corrected if necessary. Corrections were provided until the policy was successfully executed with the new object, after which an evaluation of the performance was performed. Subsequently, a different object was swapped in and the learned policy was *reset to the initial policy*.

For each new object, the policy could be successfully corrected. For the same object but with a greater weight ② the initial policy carried out the policy successfully in the first execution, hence it was deemed that no corrections were necessary. For the flexible object ③ due to its lighter weight and ease at which it could be knocked over, minor corrections to both the velocity and gripper had to be given. Lastly, for the deformable object ④ it was necessary to reduce the speed for a successful picking. Otherwise, the object kept being knocked over upon impact due to its smaller support polygon. Nevertheless, for all three objects with their different properties, it was possible to alter the policy within less time than what is needed for training from a new demonstration (see Tab. 5.2).

It is important to note that the strategies for the separate objects are not stored as this would require a further form of knowledge representation or policy parametrization, which is outside the scope of this work. This evaluation does, however, show that adapting an existing policy is faster than learning from scratch, which can be beneficial for gathering knowledge more quickly.

5.4.4 Generalizing to New Object Positions

To validate this extension we performed a short experiment where we trained the two policies (w.r.t. the object and w.r.t. the goal) with the frames fixed in one position. After the policy was successfully trained we placed the object in 20 different locations. The distance of these positions from the training location were taken from the ranges $x \in [-0.26; 0.02]$,

	T1: With Attractor Scaling				T2: Without Attractor Scaling			
	Demo Time [s]	**Training Time** [s]	**Rounds**	**Exec. Time** [s]	**Demo Time** [s]	**Training Time** [s]	**Rounds**	**Exec. Time** [s]
Max	34.10	600.00	36.00	4.97	14.90	285.00	23.00	4.00
Mean	13.04	323.30	19.40	3.42	8.63	121.22	9.11	2.81
Min	6.40	129.00	6.00	2.17	3.90	0.00	0.00	2.07

Table 5.3: Performance of Non-Experts Who Successfully Finished the Task.

$y \in [-0.30; 0.28]$, and $z \in [0; 0.08]$ all while considering locations physically feasible for the robot.

The total training time amounted to 99.4 s of which 78.9 s were needed for the corrections. Out of the 20 executions, 13 were successful without any external influence, and 3 were successful once the human physically guided the robot into the region of certainty. For the latter 3, this was in fact a desired behavior and a design choice to ensure that the robot does not generalize and potentially behave in an unsafe manner in situations it has never seen. If a person wants to add information on how to behave in these areas, this can be done by adding new points as was addressed in [55], but this was not the focus of the proposed method. The remaining 4 executions resulted in clear failure. Out of these, 2 were in the case where the object was placed at a greater height than the demonstration. After successfully picking up the object, the robot proceeded to get stuck against the surface of the table since the policy w.r.t. the goal dictated that it should be following a trajectory that was below its current position.

5.4.5 Are humans great teachers? A User Study

Since the aim of the proposed method is to enable people, who may not have a background in robotics and machine learning, to teach a robot, a preliminary user validation study was carried out. A total of ten participants aged 23 to 28 took part in this study (approved by TU Delft HREC). The same setup as in Fig. 5.5 was used, with the bag being replaced by a small square tower to provide a clearer goal. Half an hour of familiarisation with the setup was given before the actual trials began. There were two trials of ten minutes which were presented in a *randomized* order. In one trial (*T1*), users were required to perform a kinesthetic demonstration at a speed they were comfortable with. Afterwards, they had the possibility to correct the demonstration with the possibility to scale the attractor distance. To ensure that the main contribution to the velocity resulted from the scaling factor, the attractor $\Delta\mathbf{x}$ itself was bounded to 4 cm. In the other trial (*T2*), users were required to provide a *fast* kinesthetic demonstration. The attractor for this trial was left unbounded and any corrections for the velocity had to be performed by directly altering the attractor in the three Cartesian directions. *A trial was considered successful if the final trajectory execution time was* 4 s *or less*. The goal of this study was two-fold; i) verifying the feasibility of allowing non-experts to teach the robot non-zero-velocity P&P and ii) determining which correction approach users may prefer. In terms of performance, all participants were able to successfully pick & place the object in T1. Only one was unable to reach the 4 s goal. For T2, only one was unable to teach the task successfully.

Nevertheless, overall good teaching performance could be observed in both trials. For

T1, users were able to teach the task within, on average, 5.4 min with 19 correction rounds. The average time at which the robot could successfully pick & place the object was 3.4 s with the *best time being* 2.2 s. For reference, the time needed to demonstrate the behavior at a fast pace in T2 was at best 3.9 s, but generally participants needed more than 5 s to carry out the demonstrations (see Table 5.3 for detailed results). *It thus becomes clear that overall non-experts are not able to or are not comfortable with providing fast demonstrations*. Provided a faster demonstration, the time needed for corrections however did tend to be lower.

Participants were also asked which correction approach they preferred (T1 or T2). Within the group of participants, there was no clear preference for one method or the other. There were, however, clear personal preferences. Half preferred to correct the complete translational dynamics with one input, claiming that it made it easier for trajectory shaping or more intuitive for altering the velocity since it compared more closely to the controls that are familiar from video games. Meanwhile, the rest found it easier to focus on correcting one aspect at a time, thus preferring to first correct the trajectory before increasing the velocity with the scaling factor γ, since there was less chance of accidentally affecting the other aspect with the corrections. This means that by opting for only one correction approach, the performance and comfort of some people would be hampered. For this reason, it is important that the method gives people the possibility of using either of the two approaches.

5.5 Conclusions and Future Work

We demonstrated that the motion dynamics of a user's demonstration can be successfully altered in a non-uniform manner using teleoperated corrections. This allows users to overcome the limitations they had during the demonstration and teach the actual desired behavior. It further allows users to compensate for delays within the system which are not directly known to them but are observable in the system's performance. Additionally, generalization to different object positions was obtained by switching between the two dynamical systems, learned in the respective reference frames. This proved how variance minimization can be successfully used also to transition between two different frames. This opens many possibilities for creating a sequence of multiple simpler dynamical systems for accomplishing complex robot tasks, i.e., assembling multiple movement primitives.

It was additionally shown that non-experts are able to successfully teach a non-zero-velocity motion for picking & placing objects. Irrespective of their prior experience or lack thereof with robots, they were able to successfully train this complex task, teaching and correcting the motion dynamics of many degrees of freedom. It could be seen that when only using the kinesthetic demonstration, people generally could not attain the desired execution time even with a fast demonstration. However, with the help of corrections to the motion dynamics, an execution speed outside of their demonstration capabilities became achievable. Since people have different preferences for teaching and correcting robots, we concluded that the final framework requires the velocity corrections to be provided both in a coupled (with only $\Delta \boldsymbol{x}$) and decoupled manner (with γ and bounded $\Delta \boldsymbol{x}$).

Certain aspects remain to be addressed for better motion generalization, as we will see in Ch. 7. The next step would be to study how to obtain haptic corrections of the policy while ensuring a fast but safe human-robot interaction while learning long-horizon motions of single and dual arm tasks.

6

Interactive Imitation Learning of Bimanual Movement Primitives

6

Performing bimanual tasks with dual robotic setups can drastically increase the impact on industrial and daily life applications. However, performing a bimanual task brings many challenges, like synchronization and coordination of the single-arm policies. This chapter proposes the Safe, Interactive Movement Primitives Learning (SIMPLe) algorithm, to teach and correct single or dual arm impedance policies directly from human kinesthetic demonstrations. Moreover, it proposes a novel graph encoding of the policy based on GPs where the single-arm motion is guaranteed to converge close to the trajectory and then toward the demonstrated goal. Regulation of the robot stiffness according to the epistemic uncertainty of the policy allows for easily reshaping the motion with human feedback and/or adapting to external perturbations. We tested the SIMPLe algorithm on a real dual-arm setup where the teacher gave separate single-arm demonstrations. We then successfully synchronized them only using kinesthetic feedback or the original bimanual demonstration was locally reshaped to pick a box at a different height. A video of the experiments can be found at `https://youtu.be/GasxgbJZHdQ`.

This chapter is based on ***Franzese, G.****, de Souza Rosa, L., Verburg, T., Peternel, L. and Kober, J., 2023. Interactive imitation learning of bimanual movement primitives. IEEE/ASME Transactions on Mechatronics* [56].

6.1 Introduction

Modern society is faced with the lack of workforce in various repetitive jobs like re-shelving products in supermarkets or handling heavy luggage in airports. Robots appear to be the most promising solution to mitigate the negative effects of the declining workforce and perform these various complex tasks [60]. To work in variable and unstructured environments, robots must be dexterous and intelligent to quickly learn the job while interacting safely with other robots, objects, and humans. However, traditional task-specific robot programming by experts fails to achieve such dexterity and intelligence due to the time-consuming process and poor adaptability of tailored solutions.

While tasks that require only one arm have been explored extensively in the literature, more complex tasks that require a bimanual setup have only recently been targeted. Among such tasks, picking large objects in unstructured environments [60], assisting the elderly [119, 182], surgery tasks [17] or complex assembly tasks [178] are shown to require dexterous bimanual setups. Factory assembly, logistics, and household applications of bimanual robots have been known for decades [152, 176].

However, the increased number of Degrees of Freedom (DoF) (the curse of dimensionality) implies an increased teaching complexity and the necessity of skilled human teachers who know how to interface with the bimanual robotic platform.

In this chapter we contribute with the SIMPLe algorithm and propose:

- The design of a bimanual impedance controller with variable Cartesian stiffness; safety constraints on the maximum applicable force and execution velocity are also formulated;
- A novel movement primitive formulation that allows efficiently learning long horizon tasks from a single demonstration and executes the motion in a reactive way;
- Efficient corrections of the robot's policy directly from kinesthetic feedback, allowing for fine-tuning the demonstrations. Thanks to this, the user can show single arms' trajectories and fine-tune them when transferring the policies onto a bimanual task.

To validate the proposed method, we conducted a series of experiments. The first three are technical experiments related to the main contributions that highlight and test different functionalities of the method. The last two are supplementary user studies to evaluate the type of data input for the proposed by comparing two human demonstration approaches and to evaluate giving corrections compared to giving new demonstrations. These additional insights can provide a better understanding of the input data generation method and adjustments of the robot's skill for bimanual cases.

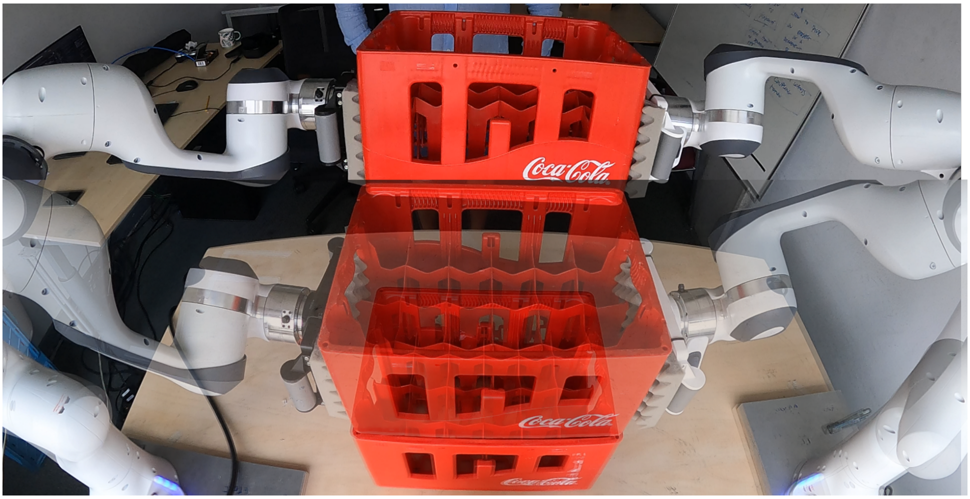


Figure 6.1: Example of possible application of bimanual manipulation: performing stacking of crates.

6.2 Related Works

6.2.1 Bi-Manual Teaching Frameworks

Like with single-arms, pre-planning and manual coding of multi-arm manipulation is a tedious process. An alternative is learning from *human demonstrations*, where a user can guide the robot on how to execute the desired tasks. However, when the user controls the dual (or multiple) robot setup, the physical and cognitive load increases drastically. Using priors, shared control or task scaffolding, i.e., dividing the teaching into smaller parts, can substantially decrease the demonstrator workload and make the teaching easier and the learning faster.

Recent works on the control side of bimanual manipulation leverage *shared control* strategies for reducing the burden of *teleoperated* bimanual tasks. For example, [99] proposes a shared controller for helping the user to perform bimanual manipulation: it maintains the manipulators' relative position (or orientation) while the user controls the translations or rotations. Similarly, [134] classifies human demonstrations in four teaching modalities: self hand-over, one-hand fixed, one-hand seeking, and fixed offset; when performing teleoperation, a trained classifier detects the most likely modality and adapts the constraints of the bimanual controller accordingly.

On the side of *shared control*, [162] extends the Roboturk platform by teleoperating each arm by a different teacher, reducing the cognitive load and enabling teaching tasks with more than two arms. Moreover, ongoing research [171] presents a controller that enables inputs from a teleoperating user and local kinesthetic perturbations. This chapter focuses on teaching bimanual policies from a single human teacher by teaching single-arm policies independently and then interactively reshaping them for successful coordination or adaptation to a new scenario. The goal is to enable non-expert users to teach complex bimanual tasks.

6.2.2 Bimanual Coordination Policies

During autonomous execution, disturbing one of the arms in a detached bimanual system can break the synchrony of the movements, making it necessary to provide both movement recovery and re-synchronization capabilities. The way the policy is encoded, e.g., time-dependent vs position-dependent, or the chosen function approximation, e.g., a Dynamic Movement Primitives (DMPs), Hidden Markov Model (HMM), GP [172] can change the disturbance rejection of the robot.

To this end, the method in [65] uses a prior on the relative position of the two manipulators and a timing dependence in the HMM formulation to synchronize the movement of arm manipulators. Other approaches propose to create a "leader and follower" movement by adding a coupling term [179], a regulation term [180], or a deterministic encoding of trajectories with DMPs [146]. Alternatively, the epistemic uncertainty of GPs can be used for switching the behavior of the arms from follower to leader (and vice-versa) [49]. This leader-follower learning paradigm makes the system react differently according to which arm is perturbed. Alternatively, symmetry prior can be used to easily encode and synchronize the task. For example, [23] proposes a bi-manual policy for picking and throwing non-stationary objects by learning a *symmetric* dynamical system policy. In this case, perturbing any of the two arms would always make the other react.

Other approaches focus on achieving synchrony and coordination by segmenting the trajectories and reproducing them in sequence or according to a *hierarchical* representation of the task. The advantage of such approaches is that the sequencing provides an implicit synchronization on a higher level, making the lower-level problem easier. A common approach for this scheme is to learn policies for performing pre-defined sub-tasks, and a higher-level policy which creates a sequence from demonstrations [96, 112]. Alternatively, the task can have a pre-defined structure of sub-tasks based on heuristics, and synchrony is achieved with a sub-task scheduler [27]. Segmentation has also been used for deep-learning bimanual tasks in [175], where lower-level policies are learned for each segment and higher ones for sequencing them. In this direction, [144] proposes a framework for multi-arm task-space control with smooth transitions from independent behaviors, e.g., when reaching goals, to dependent ones, e.g., when performing a dual-arm manipulation.

Our proposed approach differs from the approaches mentioned above in two ways. First, these approaches fall under the LfD category while our proposed SIMPLe framework is an IIL algorithm, and to the best of our knowledge, SIMPLe is the first framework for learning of bimanual tasks from interactive corrections. Second, our interactive framework avoids heuristics for coordinating policies for each arm in a bimanual setup by using human feedback to regulate each arm's dynamics before transferring it to a bimanual policy. Then, when the bimanual policy is executed, the robot's reaction to disturbances depends on the mechanical coupling of the end-effectors (see Appendix A.2) or on the chosen input state for the policy (see Section 6.3).

6.2.3 Motion Stability

The stability of the bimanual operation is another key aspect. When learning from a small amount of data, in particular, the stability of the learned behavior can be jeopardized when demonstrations are imperfect. In [59, 106], a LfD approach is combined with a learned controller that adapts the motion to keep the learned trajectory stable when facing external forces. In [23], the motion is divided into one Dynamical System (DS) for each sub-goal with a hand-designed vector field that brings the robot always close to the connecting lines of sub-goals. Our proposed Movement Primitives (MPs) have the objective of learning long-horizon MPs with only one final goal and to obtain the stability property as an emerging behavior of the motion encoding (Section 6.3.3).

Next, Section 6.3 introduces the novel GP-based formulation used for modeling MPs, Section 6.4 introduces the proposed SIMPLe algorithm and how we use it for performing interactively learning bimanual MPs, Section 6.5 shows different applications and user-cases, and Section 6.6 concludes the article with final remarks and future works.

6.3 Movement Representation

Section 6.3.1 presents the proposed Graph Gaussian Process (GGP) formulation, Section 6.3.2 the proposed trajectory learning framework and its benefits for safety, Section 6.3.3 presents the stability achieved with the proposed framework, and Sections 6.3.4 and 6.3.5 compare learning trajectories using traditional GPs and the proposed GGPs.

6.3.1 Movement Learning with Gaussian Process

To learn the model of the demonstrated trajectories, we chose GPs because it is a flexible non-parametric regression method where the kernel choice can be used to increase the inductive bias on the generalization of unseen states, which is prohibitive using function approximators such as DMPs or Neural Networks (NNs). Furthermore, its solid statistical formulation provides both the mean and the epistemic uncertainty of the prediction [172] that can be used for disturbance rejection or stiffness regulation, as in Ch. 4 or Ch.5.

In particular, the kernel determines the interpolation and extrapolation behaviors and when using a distance-based kernel, i.e. Squared Exponential kernels, the prediction converges to the mean of the Gaussian Process, usually set to zero. Our objective is to have a mean function that extrapolates without losing the measure of epistemic uncertainty, i.e., does not return a vanishing prediction. For this reason, by correlating with only the closest neighbor in the dataset, the kernel definition becomes:

$$\tilde{\boldsymbol{k}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \begin{cases} 1, & \text{if } \boldsymbol{k}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \max(\boldsymbol{k}(\boldsymbol{x}_i,\boldsymbol{X})) \\ 0, & \text{otherwise} \end{cases} \forall \boldsymbol{x}_j \in \boldsymbol{X}.$$

In simple terms, given a point $\boldsymbol{x}_i$, the correlation is 1 only if that is the maximum obtainable correlation when correlating $\boldsymbol{x}_i$ with all $\boldsymbol{x}_j \in \boldsymbol{X}$. With the new kernel, the prior covariance matrix becomes:

$$\Sigma_{prior} = \begin{bmatrix} \tilde{\boldsymbol{K}} & \boldsymbol{k}_\star \\ \tilde{\boldsymbol{k}}_\star^\top & \boldsymbol{k} \end{bmatrix}.$$

Note that, since the last column is for $\boldsymbol{x}_j \notin \boldsymbol{X}$, the saturation is not applied. Thus, the resulting prior covariance matrix is no longer symmetric, making the new process a pseudo-GP. After the conditioning on the data points, the new pseudo-GP posterior becomes:

$$\mu(\boldsymbol{x}) = \tilde{\boldsymbol{k}}_\star^\top \tilde{\boldsymbol{K}}^{-1} \boldsymbol{y} = \tilde{\boldsymbol{k}}_\star^\top \boldsymbol{y},$$

$$\sigma^2(\boldsymbol{x}) = \boldsymbol{k} - \tilde{\boldsymbol{k}}_\star^\top \boldsymbol{k}_\star.$$

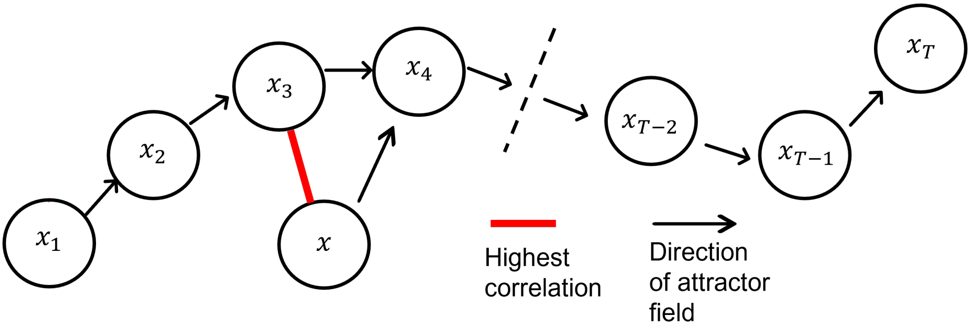


Figure 6.2: Representation of a trajectory as a chain of events. The state $\boldsymbol{x}$ is the aggregation of the robot pose and time, where $\boldsymbol{x}_i$ is i-th element in the reference trajectory. Every element of the trajectory i-th has as its goal the $(i+1)$-th element depicted by a forward arrow. The $\boldsymbol{x}$ gets a unitary correlation with the closest element in the trajectory m-th and then as the goal the $(m+1)$-th state on the chain of events. The uncertainty is given by the distance between the $\boldsymbol{x}$ and its correlated point on the trajectory.

In simple terms, $\tilde{\boldsymbol{k}}_\star$ selects as mean the label of the closest point in the database, computing the uncertainty according to the relative position between the query and the selected points.

Additionally, by saturating the covariance matrix $\boldsymbol{K}$, each trajectory element has its highest correlation with themselves: the new saturated correlation matrix, $\tilde{\boldsymbol{K}}$, is the identity matrix, thus eliminating the computationally heavy $\mathcal{O}\left(n^3\right)$ matrix inversion. However, with this approximation, we are losing interpolation/smoothing properties. Meaning that the provided trajectory data must be without drastic jumps. In practice, recording trajectories with high enough frequency (> 10Hz) and/or smoothing the data makes the use of the proposed approximation doable. It is worth mentioning that the presented formulation is tailored for the specific application of movement learning and does not necessarily substitute general approximation methods like local models [148] or variational approximations [72]. A detailed comparison between GPs and GGPs for trajectory learning is presented in Section 6.3.4.

6.3.2 Representing Trajectories as Graphs

Our goal is to perform safe control during the general or corrective interactions between robots and humans. To that goal, we start from a recorded trajectory demonstration, defined as an array of n end-effector poses $\boldsymbol{\xi} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}\} \in \mathbb{R}^3$ and the timestamp of each respective pose $\boldsymbol{\tau} = \{t_0, \ldots, t_{n-1}\} \in \mathbb{R}$, and a final pose and time $\boldsymbol{x}_n, t_n$, used to fit a policy π. The trajectory can be seen as a sequence of events, represented as a graph with edges representing transitions from the state at time t_i to the state at t_{i+1}. Given the adopted GP approximation, during the policy execution, the most correlated point is selected on the trajectory, and its label is selected as the goal, see Fig. 6.2. We denote the policy as a GGP.

However, the input type of the policy can completely change the robot behavior. For example, a pose-only "feedback" policy, $\pi_{\boldsymbol{x}} : \boldsymbol{x} \rightarrow \boldsymbol{x}_g$ is a fully reactive policy which computes the next Cartesian pose for the end-effector $\left(\boldsymbol{x}_g\right)$, based on the current one $(\boldsymbol{x})$. Such policies are safer since they make the robot wait when its path is obstructed and allow it to rejoin the trajectory on its closest point under perturbations [55]. However, they cannot deal with movement ambiguities and time-dependent movements.

Alternatively, a time-only dependent policy, $\pi_t : t \to \boldsymbol{x}_g$, computes $\left(\boldsymbol{x}_g\right)$ based on the current time (t). This type of policy can deal with movement ambiguities, e.g., when the demonstrated trajectory crosses itself, and with time-dependent movements, i.e., when the movement has to be temporarily paused at a specific position. However, such "feed-forward" policies are not a safe choice since the attractor moves on the trajectory without considering dangerous interactions with humans and with the environment.

Instead, we proposed the usage of pose and time-*belief* dependent policies, $\pi_{\boldsymbol{x},t^b} : \boldsymbol{x}, t^b \to \boldsymbol{x}_g, t^b_g$, which computes the pose goal and a new time belief $\left(\boldsymbol{x}_g, t^b_g\right)$ based on the current ones $\left(\boldsymbol{x}, t^b\right)$. Note that the time-*belief* is updated with the time of the selected goal in the trajectory. Encoding both pose and time belief allows for obtaining safe policies capable of handling time-dependent movements and ambiguities.

As such, SIMPLe can be used with models fitted as time-dependent, pose-dependent, or pose and time-dependent policies by setting the GGP states as $\boldsymbol{x} := t$, $\boldsymbol{x} := \boldsymbol{x}$, or $\boldsymbol{x} := \left[\boldsymbol{x}, t^b\right]^\top$, respectively, and selecting a kernel for fitting the trajectories w.r.t. time $(\boldsymbol{k}(t,\boldsymbol{\tau}))$, like in [77], position $(\boldsymbol{k}(\boldsymbol{x},\boldsymbol{\xi}))$, like in [55], or both of them, as proposed in SIMPLe, which is obtained by multiplying the time and the pose-dependent kernels, i.e., $\boldsymbol{k}\left(\left[\boldsymbol{x}, t^b\right], [\boldsymbol{\xi}, \boldsymbol{\tau}]\right) = \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{\xi}) \circ \boldsymbol{k}(t, \boldsymbol{\tau})$.

In the context of trajectory learning, the labels are set as the aggregation states in the demonstration which follow each state in the demonstration, i.e., $\boldsymbol{y} = \left[\boldsymbol{\xi}^d, \boldsymbol{\tau}^d\right]^\top = [\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \{t_1, \ldots, t_n\}]^\top$.

6.3.3 Stability Analysis

From this GGP-based formulation, we can also conclude that:

Proposition 6.3.1 *Using the trajectory graph representation, the motion always converges on the proximity of the demonstration and continues towards the end of it.*

Proof 6.3.1 *Since the vector $\tilde{\boldsymbol{k}}_\star^\top$ is correlating the current position of the end-effector with only one node of the trajectory, and if there is no overlap on the trajectory, the robot will move towards the goal of the closest node. Then, node by node, it continues toward the end of the trajectory.*

A great advantage of the pose and time trajectory encoding is that overlapping is no longer possible as the demonstrator cannot show two different robot positions simultaneously, leading to the absence of overlapping nodes, ambiguities, or undesired loops, guaranteeing that the hypothesis in the proof of convergence is satisfied. However, this also means that when *only* computing the correlation as a function of position, no physical overlapping of the trajectory can be demonstrated, such as when drawing an eight [164].

6.3.4 Comparison Between GPs and GGPs for Policy Learning

Figure 6.3 shows the different behavior in learning to draw the letter "B" (database from [77]) using a GP and a GGP using only the 2-D position. The first thing to highlight is the

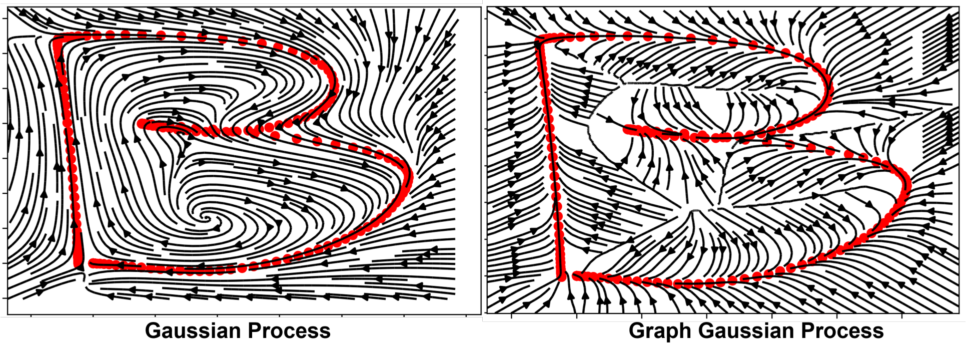


Figure 6.3: Comparison of the fitting of a trajectory with the shape of a “B” with *position dependent* GP and GGP. The red dots are the recorded demonstrations, and the stream curves are the learned behavior.

effect of the kernel saturation in a faster convergence closer to the trajectory of the GGP compared with the GP. As consequence, when the robot is perturbed, the motion tends to go closer to the trajectory and continue from there. Nevertheless, this difference in the vector fields does not lead to unsafe sudden motions straight towards the attractor due to the proposed attractor and stiffness regularization/saturation described in Appendix A.1.

The letter “B” shows a clear ambiguity at the overlapping of the trajectory between the two humps. The robot must first move in and then move out of the intersection on the same line in order to continue towards the end of the trajectory. The learned behavior of the two fitting methods is different. The GP removes the overlapping ambiguity by considering it as noise. This results in cutting the motion without going down to the intersection of the curves, losing tracking accuracy. On the other hand, in the line overlapping, the GGP has a vector field pointing left when approaching from below and to the right when approaching from the top. This may lead to an ambiguous situation that can cause the robot to get stuck locally or, in general, not track the motion correctly. This motivates the use of a position and time-dependent policy, to remove any possible state overlapping.

6.3.5 Movement Disambiguation using Pose and Time-Dependent Policy

As explained in Proposition 6.3.1, no loops in the chain are allowed to guarantee good trajectory tracking. Thus, our solution is to consider also the time belief $\left(t^b\right)$ in the state. Figure 6.4 shows the evolution of the vector field for different time beliefs. The chain element of the trajectory for the t^b indicated above the figure is highlighted with a green dot. From the figure, it is possible to observe how the previously encountered ambiguity is elegantly solved. In fact, the robot gets into the valley and then out without getting stuck.

In order to simulate the behaviors of a GGP with or without a self-update of the time belief, 200 different trajectories are rolled out starting from the origin of the demonstration. In order to take into account the inaccuracy of the low-level (impedance) controller, a Gaussian noise of magnitude 0.01 is added to the attractor when computing the new position. Figure 6.5 depicts the mean and standard deviation of the trajectories. When the time dependence is active (left side of the figure), the trajectory always converges to the end, and

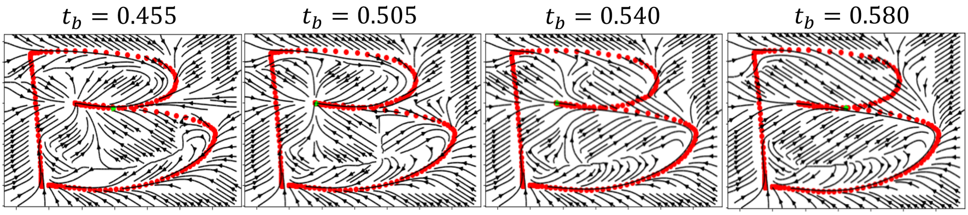


Figure 6.4: Attractor vector-field of a Graph Gaussian Process when the conditioning of the kernel with different time belief. t_b is the time belief normalized by the total time of the trajectory, i.e., $0 \leq t_b \leq 1$. The corresponding element of the chain for every case is highlighted with a green circle. The red circles are the collection of points of the trajectory. In this example, the time belief is the time of the next element of the chain.

the fluctuations are bounded. When only the position is considered (right side of the figure), the variability of the sampled trajectories increases, and the tracking is good on average until the start of the two humps intersection, from where the performance degrades due to the ambiguous states.

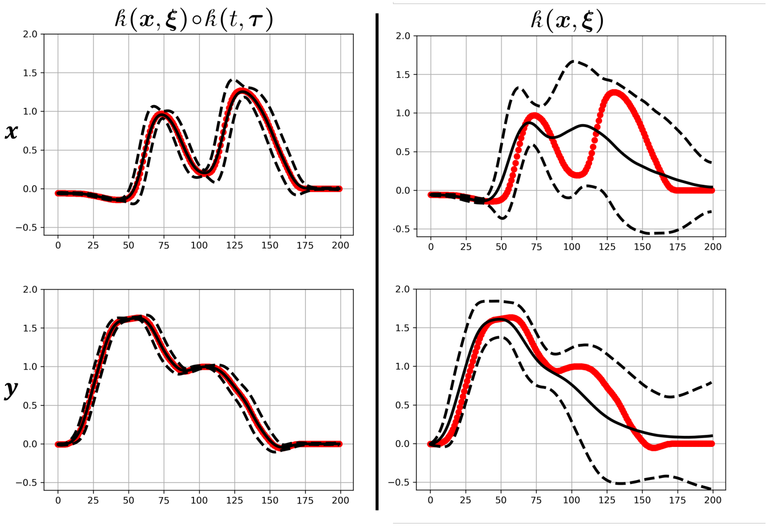


Figure 6.5: 200 roll-outs of 200 steps. The attractor position is injected with a Gaussian noise of zero mean and std of 0.01. When the kernel is also taking into account the time belief, the motion is more robust when encountering ambiguity in the intersections of the two curves. Otherwise, ambiguity can lead to divergent behaviors. Red: original demonstration. Black: the average and standard deviation of the execution.

6.4 SIMPLe: Safe, Interactive Movement Primitives Learning

The proposed SIMPLe framework summarized in Algorithm 3 consists of three main parts. First, the human teacher provides kinesthetic demonstrations (Section 6.4.1), from which a time and position-dependent model (Section 6.3) is learned. Second, the proposed method enables the human to provide demonstrations and to make interactive corrections (Section 6.4.1), which are leveraged for learning the trajectories and synchronization of bimanual tasks (Section 6.4.2). And third, the bimanual task can be executed. We employ a Cartesian impedance control to facilitate physical interactions during demonstrations, corrections and autonomous execution (Appendix A.1), safety is ensured thanks to the proposed stiffness regulation (Section 6.4.3) and coupling between manipulators (Appendix A.2).

Our method aims to enhance the teaching ability of non-expert users while guaranteeing a safe interaction while teaching, correcting, and executing bimanual tasks. To cope with the complexity of teaching bimanual tasks, SIMPLe provides an interactive kinesthetic teaching (KT) approach allowing to teach one arm at a time and then to teach how to synchronize them using touch by leveraging the time and pose-dependent GGP formulation presented in Section 6.3. To the best of current knowledge, SIMPLe is the first framework to employ IIL on bimanual setups. Nevertheless, SIMPLe does not restrict users from teaching (and correcting) both arms simultaneously, and it can be applied for single-arm manipulation tasks without any loss of generality.

6.4.1 TEACHING FROM KINESTHETIC DEMONSTRATIONS AND CORRECTIONS

LfD allows non-expert users to program robots to perform complex tasks without any programming knowledge. Different interfaces can be used to transfer data to the robot, such as teleoperation devices, touch screens or physical interaction with the robot's embodiment, obtaining a KT approach. When the user is teaching a task, the stiffness and damping of the Cartesian impedance controller are set to zero, allowing the user to easily move the robot. The positions $\boldsymbol{\xi}$ and times $\boldsymbol{\tau}$ of the demonstrated trajectories are recorded, and their respective goals, $\boldsymbol{\xi}^d$ and $\boldsymbol{\tau}^d$ are obtained by shifting $\boldsymbol{\xi}$ and $\boldsymbol{\tau}$ forward in time (Alg. 3, lines 1 to 5).

After learning the motion from a kinesthetic demonstration, the user can reshape the trajectory of each arm to achieve, for example, coordination between the arms in the execution of the task. Given the Cartesian impedance controller (see Appendix A.1), kinesthetic corrections can be performed by simply applying an external force . Such a controller allows for the human to be in full control if the stiffness is set to zero, or the robot can gradually increase its control by regulating the stiffness.

Additionally, given the time and pose-dependent policy (see Sec. 6.3), the demonstrator can also drag the robot forward or backward in time along its trajectory. This property can be used, for example, to make the execution of the initial demonstration faster [86, 116], to make the robot throw objects [23], or for synchronization learning, as proposed in this chapter.

6.4.2 INTERACTIVE LEARNING OF BIMANUAL TASKS

When teaching bimanual tasks, it is not always easy or feasible to provide kinesthetic demonstrations with both arms simultaneously, especially when using large redundant manipulators. Additionally, even when skilled users are able to teach a bimanual task by moving each end-effector with a single hand, they may perform a sub-optimal trajectory, or an ineffective one, given the task complexity.

Algorithm 3 SIMPLe

```
while Trajectory Recording do
    $[\boldsymbol{x}, t]^\top = \text{Receive}(\boldsymbol{x}, t)$
    $[\boldsymbol{x}, t]^\top \uplus \boldsymbol{X}$
end while
$\boldsymbol{y}_i = \boldsymbol{X}_{i+1}, \forall i \in \{1, 2, \ldots, n-1\}$, with $n = \dim(\boldsymbol{X})$
$t^b = 0$
while Interactive Corrections do
    $[\boldsymbol{x}, t]^\top = \text{Receive}(\boldsymbol{x}, t)$
    $[\boldsymbol{\mu}, \boldsymbol{\sigma}] = \pi_{\boldsymbol{x}, t^b}(\boldsymbol{x}, t^b)$
    $[\boldsymbol{x}_g, t_g^b]^\top = \boldsymbol{\mu}$
    $t^b = t_g^b$
    Send(saturate($\Delta\boldsymbol{x}$), $\hat{\boldsymbol{\mathcal{K}}}$)
    $[\boldsymbol{x}, t]^\top \uplus \boldsymbol{X}$
end while
$\boldsymbol{y}_i = \boldsymbol{X}_{i+1}, \forall i \in \{1, 2, \ldots, n-1\}$, with $n = \dim(\boldsymbol{X})$
$t^b = 0$
while Autonomous Execution do
    Receive($\boldsymbol{x}$)
    $[\boldsymbol{\mu}, \boldsymbol{\sigma}] = \pi_{\boldsymbol{x}, t^b}(\boldsymbol{x}, t^b)$
    $[\boldsymbol{x}_g, t_g^b]^\top = \boldsymbol{\mu}$
    $t^b = t_g^b$
    Send(saturate($\Delta\boldsymbol{x}$), $\hat{\boldsymbol{\mathcal{K}}}$)
end while
```

In SIMPLe, the movement of each arm can be executed independently according to the GGP formulation described in Section 6.3. The proposed interactive learning method offers many possibilities for non-expert users to teach complex bimanual tasks. For example, they can demonstrate the movement for picking up a box one arm at a time and then learn to coordinate the two independent trajectories and apply enough pressure on the sides of the box to execute the task successfully. Moreover, learning repetitive tasks like object hand-over can also be initially demonstrated one arm at a time and later use kinesthetic corrections to learn how to coordinate both arms. Thanks to the calculation of the model as a function of position and time (belief), the user can also bring the robot back to the start of the trajectory and teach (with minimum interaction effort) to perform the task multiple times.

6.4.3 STIFFNESS REGULATION

Regulating the stiffness can be used to incrementally increase the stiffness after each demonstration, reducing human control as the learned movement is interactively refined [163]. Alternatively, the stiffness can be regulated when perceiving strong external forces, as a disagreement detection [165]. Similarly, [86] proposed a variation of a DMP where the robot variable stiffness and the regressor phase are modulated to adapt to human kinesthetic demonstrations.

When more demonstrations are provided, the measure of *aleatoric* uncertainty, i.e., variability in the demonstration, can be used to regulate the tracking stiffness of the robot [81]. Differently, we propose to exploit the *epistemic* uncertainty quantification of the policy ($\boldsymbol{\sigma}$), enabling for automatically regulating the Cartesian impedance controller's stiffness, hence switching control between robot and human.

Mathematically,

$$\hat{\boldsymbol{\mathcal{K}}} = \text{saturate}(\mathcal{K})\frac{1-\boldsymbol{\sigma}(\boldsymbol{x})}{1-\boldsymbol{\sigma}_{tr}}, \text{ when } \boldsymbol{\sigma}(\boldsymbol{x}) > \boldsymbol{\sigma}_{tr} \tag{6.1}$$

where the $\boldsymbol{\sigma}_{tr}$ is the uncertainty threshold that is used to detect the disagreement. Note that $\boldsymbol{\sigma}(\boldsymbol{x})$ goes from 0 when close to the trajectory, to 1 when at infinite distance from it. Thanks to this stiffness regulation, when the robot is dragged in regions of high uncertainties, it mitigates the external force applied to the user perturbing the trajectory. This behavior can be conceptualized as the robot's non-verbal teaching request or repositioning into regions closer to the demonstration.

6.5 REAL ROBOT VALIDATION

We performed the experiments with two 7-DoF Franka-Emika Panda placed vertically on a table and with the same orientation. The impedance control was implemented[1] as described in Appendix A.2. Each manipulator had a shared memory of their Cartesian poses, allowing the calculation of the mechanical coupling force. The experiments presented in Sections 6.5.1, 6.5.2, 6.5.4, and 6.5.5 were performed using a custom 3D-printed plate end-effector depicted in Figs. 6.9 and 6.10, which features a layer of soft form for reducing the interaction forces during impacts with objects as in [44]; the experiment presented in Section 6.5.3

[1] https://github.com/franzesegiovanni/franka_bimanual_controllers

was performed using the Franka gripper. The impedance control framework, written in C++ makes use of Robot Operating System (ROS) to interface with Alg. 3, written in Python. [2]

We perform 5 experiments with the real robot setup:

i) The interactive synchronization of the picking motion of a bottle crate when the demonstration is provided separately for each robot, showing how SIMPLe is used to learn a bimanual synchronization,

ii) the interactive correction in picking a different crate compared to the one of the original demonstration, showing how to use the GGP formulation to modify the motion locally,

iii) a handover task, where one robot picks and places an object and the other robot picks it from the other's goal location and places it at another position, showing the ability to restart the execution of a trajectory simply dragging the robot at the starting location,

iv) a supplementary user study to compare teleoperation and KT, the two most common types of demonstration approaches,

v) a supplementary user study to compare giving interactive corrections to giving new demonstrations.

The first three are technical experiments to highlight and validate different functionalities of the proposed method. Each experiment was conducted in 5 trials, and for each of them, the final learned motion was performed 5 times after demonstration and correction(s). This approach allowed for the assessment of the reliability of the learned skill. The last two are supplementary user studies to evaluate the type of data input for the proposed by comparing two human demonstration approaches and to evaluate giving corrections compared to giving new demonstrations. These additional insights can provide a better understanding of the input data generation method and adjustments of the robot's skill for bimanual cases. For all the experiments, we used a position-time kernel for the GGP that computes the correlations and updates the time beliefs online. We use a negative exponential kernel, i.e. $k = \exp\left(-\frac{|x_i - x_j|}{l}\right)$, with a length scale of 0.05 m for the space correlation and 0.05 s for the time correlation. The sigma threshold is set to $\sigma(l)$, which is the uncertainty when the closest point is at a distance l. The Cartesian stiffness is kept to 600 N/m for linear stiffness and 30 Nm/rad for rotational. The attractor distance is saturated at 0.05 m, implying that the expected maximum applicable force is 30 N in every linear Cartesian direction and the maximum expected linear velocity is $\approx$ 0.6 m/s in every linear direction. The rotation delta is saturated at 0.15 rad, implying a maximum torque of 4.5 Nm in every rotational component and a maximum velocity of $\approx$ 0.4 rad/s. The coupling stiffness is set to 800 N/m in the linear components and 0 for the rotational ones. The relative error is also saturated at 0.05 m. A video of the experiments can be found at https://youtu.be/GasxgbJZHdQ.

6.5.1 Asynchronous Crate Picking

When a pianist approaches studying a new piece, they do it *one hand at a time*. After mastering the movement with each hand, they start learning how to successfully coordinate

[2] https://github.com/franzesegiovanni/SIMPLe

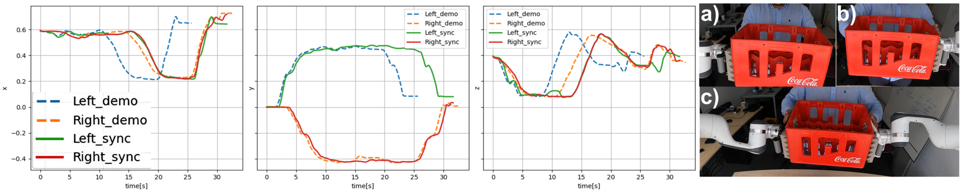


Figure 6.6: Interactive synchronization of a bimanual picking task. The dashed lines are the demonstrations recorded in the independent demonstration phase a) and b). Since they are not perfectly synchronized, the autonomous execution would fail, hence, the human feedback in c) allows a successful synchronization, depicted with solid lines.

the combined execution. Inspired by this idea, in this validation experiment, the user is asked to demonstrate how to best pick a crate, first with the right and then with the left manipulator. However, when the independently learned behaviors were executed with SIMPLe the coordination was off, and the handling of the crate was not stable. In Section 6.4.2, we highlighted how user feedback can be used to reshape the trajectory and that the reactive formulation of SIMPLe makes the trajectory to “virtually” stop: this feature can be used to learn a bimanual task while simply coordinating the separately recorded policies, see Figure 6.6.

The effect of the human input can be appreciated in Figure 6.6. The original demonstrations are represented by dashed lines. Even if the movement of the two demonstrations looks correctly symmetric with respect to the y-plane, the right arm is slower. However, it can be noticed how, after only one correction round, the motion of the two demonstrations is synchronized, as depicted with a solid line. Given the perfect obtained synchronization, in the next round, the user focused on increasing the applied pressure on the side of the crate to increase the grasp reliability.

In the 5 experiment repetitions, the user consistently provided necessary synchronization corrections. One trial had an additional correction round, and two trials had two extra correction rounds. After the interactive correction rounds, the robot always placed the crate correctly. The Cartesian error of the final crate position with respect to the final round of correction, considering 25 repetitions (5 executions x 5 trials), has a mean of 0.021 m and a standard deviation of 0.009 m.

6.5.2 Synchronous Crate Picking

In this experiment, we focused on successfully teaching the same task of picking a box but giving bimanual demonstrations and corrections. In particular, we showed that even giving only one bimanual demonstration with a few rounds of corrections, the task execution was successful. We also tested the possibility of locally modifying the original policy to pick a different box placed at a higher level. Figure 6.7 highlights how the robot can be dragged higher sooner, at around 10 seconds, and how, after picking the crate, the robots follow the original policy, being able to place the crate and go to resting position autonomously. In the 5 experiment repetitions, in the first two trials, the user provided two rounds of correction, but only one in the last three. The final position error of the box has a mean of 0.005 m and a standard deviation of 0.004 m. It is important to notice that even knowing the box’s position,

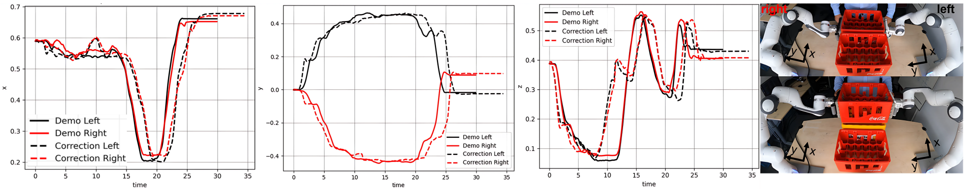


Figure 6.7: Use interactive learning to teach the robot how to modify the original trajectory so the robot can learn how to pick a crate that is at a different height.

the motion's generalization in a task-parameterized approach is not trivial. In fact, the policy would have to move with respect to the picking frame and then, after a successful pick, switch with respect to the goal crate. This logic has shown to be successfully implemented in [116] but also to be a source of generalization ambiguities [54]. In general, performing a shared controlled teaching, with the user only taking control locally, can drastically reduce the burden of giving new complete demonstrations.

6.5.3 Object Hand-over

Another example of a tedious task is repetitive demonstrations: being able to demonstrate the task only once and then interactively assemble a long trajectory allows the teaching of complex bimanual coordination tasks, like stirring a coffee mug [154] or learning a handover task. To validate SIMPLe in this circumstance, we taught the right arm to pick up a box and place it on the central separation line between the two robots. Then, the left arm would pick up the box and place it in its front. The goal is to show how dragging the robot around can be used for re-synchronization or local trajectory reshaping and also as a movement "reset".

The original demonstrations are displayed with a dashed line in Figure 6.8. When executing the motion with SIMPLe, the human can safely apply a force on the robot to stop its execution or drag it around on another desired position of the motion. At the beginning of Figure 6.8, a force is applied to the left manipulator (highlighted by a red circle) to temporally stop it from moving, allowing the right arm to successfully pick a box and place it on the center line. At the moment that the user releases the robot, it is free to move and can pick up the box and reach its goal. To allow the repetition of the motion, the user applies a larger external force (observable with peaks), causing a drop in stiffness since the robot is probably dragged into a region of space with a lower correlation according to (6.1). Every time the robot finishes its pick and place task, if the user is willing to repeat it, they only have to drag the robot to the desired position of the trajectory. The user is teaching the motion multiple times, as reported with colored patches in the figure.

We measured the final error in placing the box after the handover, executed 5 times in 5 different demonstration trials. The mean error and standard deviation are 0.011 m and 0.008 m, respectively.

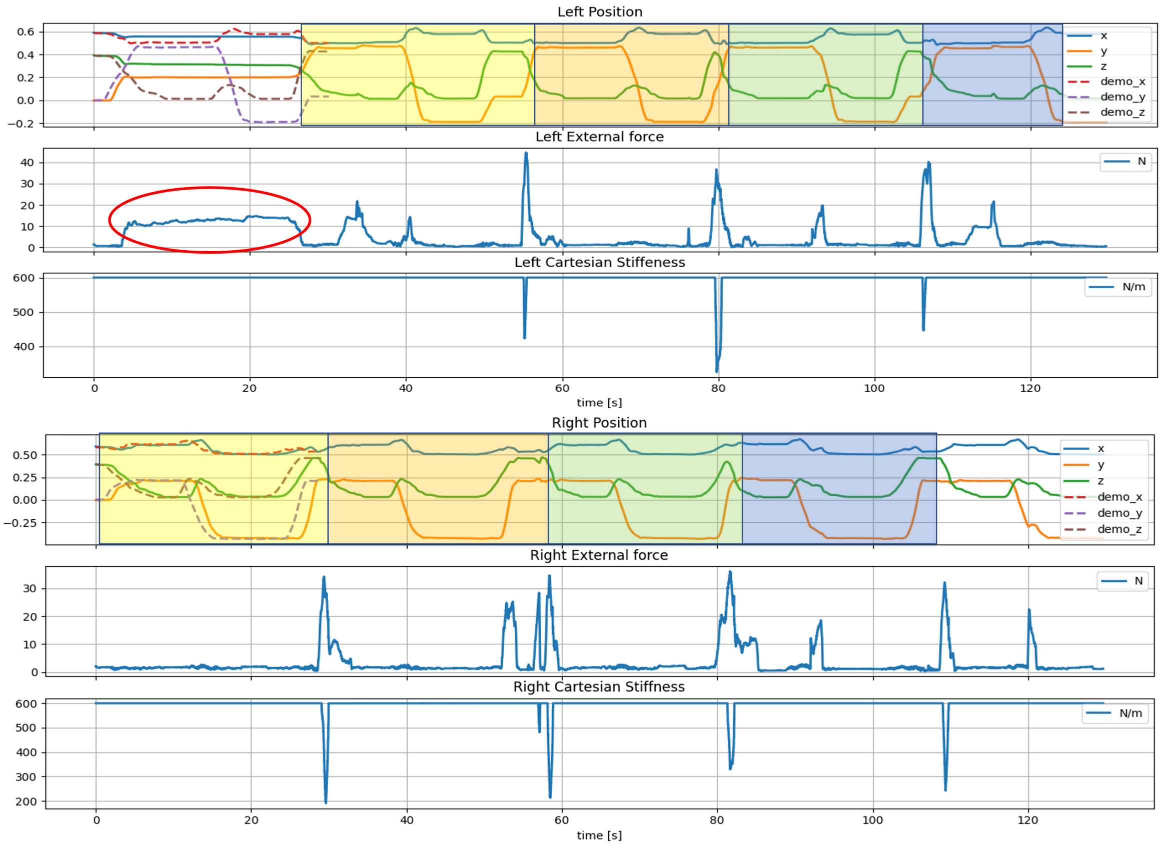


Figure 6.8: Object hand-over. For both the left and the right arm, the Cartesian position (x,y,z) is depicted as a function of time, the Cartesian linear stiffness, and the norm of the Cartesian external force. The red circle indicates the region where an external force was applied to the left manipulator to correct the trajectories.

6.5.4 User study: Teleoperation vs. Kinesthetic Teaching

The algorithm itself works with different data from different types of demonstrations. However, since obtained input data depends on the type of demonstration, the demonstration modality is an essential part of the whole framework. Therefore, we conduct a supplementary user study to provide additional insight into the effects of the demonstration method to compare the two most common demonstration approaches: teleoperation and kinesthetic guidance. There are studies comparing both teaching approaches, but they were conducted for a single arm [51, 62]. The study in this chapter looks into this subject from a bimanual perspective.

Section 6.2 highlighted how different works focus on enhancing the teleoperation ability of non-expert users using assistive techniques like shared autonomy [99, 134, 162]. Since SIMPLe works with both teleoperated and kinesthetic demonstrations/corrections, we wanted to study which is more user-friendly. Although, getting the true answer is not easy: the teleoperation device can have a strong influence, as well as the dimension of the robot or the requested task. For the conducted user study, we asked 7 non-expert users to perform a relatively simple task: pick a box and stack it on top of another. These 7 users were all male and with ages ranging 23 and 40 years old.

In order to mitigate the learning bias from the results, participants had a familiarization phase for each teaching modality, in which they could restart the teaching session up to 5 times. For every new participant, the first teaching modality was alternated between teleoperated and kinesthetic, to remove the bias due to their familiarization with the task.

For metrics, we measured the success rate in solving the task and the total teaching time for each method. For subjective analysis, we asked the participants to complete a NASA TLX questionnaire. We conducted a paired samples t-test to verify if the time to do KT is significantly shorter than for teleoperation with the 6D mice. However, 3 people out of 7 failed to perform successful teleoperation, because they did not manage to coordinate well, making the robot self-collide or reach a joint limit. Therefore, we set as failure time the maximum time of the non-failing ones. The test showed that KT requires less time compared to the teleoperation with the given hardware with the difference being statistically significant ($p < 0.05$).

Figure 6.9 illustrates the average NASA TLX scores among the different users. We can observe that teleoperation resulted in being more mentally demanding and frustrating to perform. In general, we could observe that users tend to focus on teleoperating one arm at a time, making handling the box impossible. When providing KT, the physical contact with the robot helps them to understand the best trajectory better and to accomplish the task successfully.

6.5.5 User study: Corrections vs. New Demonstration

Besides the input data generation method, another key factor related to bimanual manipulation teaching is how humans correct existing skills and what their preference is between correcting or giving a new demonstration. To test this, 12 non-expert users participated in an experiment structured as follows. The user was asked to demonstrate the task of placing a box on the crate. The demonstration was then shown to the user after an offset was applied to the initial position of the box. The user was then tasked with kinesthetically correcting the initial policy to account for the change in the initial position. This was repeated two times

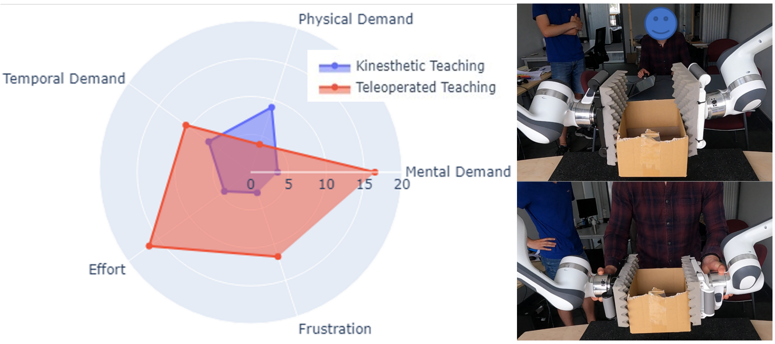


Figure 6.9: User study to compare the performance of non-expert users in performing bimanual teleoperation with two 6-D mice versus bimanual KT. On the left are scores of the NASA-TLX questionnaire, and on the right are the set-ups of teleoperating and performing KT.

for different initial positions of the box. The user now should have a sufficient understanding of what it means to give a demonstration or correction. Fig. 6.10 illustrates the setup with the box to pick. The goal is to place the box on the crate. The rectangles are the different regions where the box can be placed or dragged to as a waypoint.

The second part of the experiment was designed to find the user preference for increasing lengths of the demonstration. The user was tasked with first demonstrating the task of placing the box on the crate. After the demonstration, an offset was applied to the box and the user was given the choice to either correct or re-demonstrate given the new initial condition. For the second iteration, the task remained the same with the additional requirement that after picking up the box, before placing it on the crate, the user has to move the box through a different location as a waypoint. This was done to artificially lengthen the demonstration. Once again an offset was applied to the initial position of the box and the user was given the choice between correcting or redemonstrating. This was done one last time with two waypoints.

Given the choice, out of the 12 participants, 11, 8, and 10 chose to adjust the policy with the interactive corrections for the experiment with zero, one, and two waypoints, respectively, rather than providing new demonstrations. Thus, only in 7 out of the 36 trials, a new demonstration was preferred, which indicates a strong preference for interactive corrections. Afterwards, to evaluate their experience they were asked to answer several Likert scale questions related to user perception of corrected skill and their physical/mental load. The results can be seen in Table 6.1, where the number in each cell represents the number of participants that choose a particular agreement on the Likert scale.

The users found that both new demonstrations and corrections were effective at improving the robot's task. The users were split on whether the bimanual demonstrations were tedious. In general, they found interactive corrections more physically demanding than providing new demonstrations, probably because the robots were already performing movements rather than being completely compliant during new demonstrations. During the experiments, it was observed that people that were shorter, had smaller hands, or were less muscular, tended to struggle more with correcting a policy. Those participants thus might have preferred giving a new demonstration over a correction. However, the users perceived interactive corrections

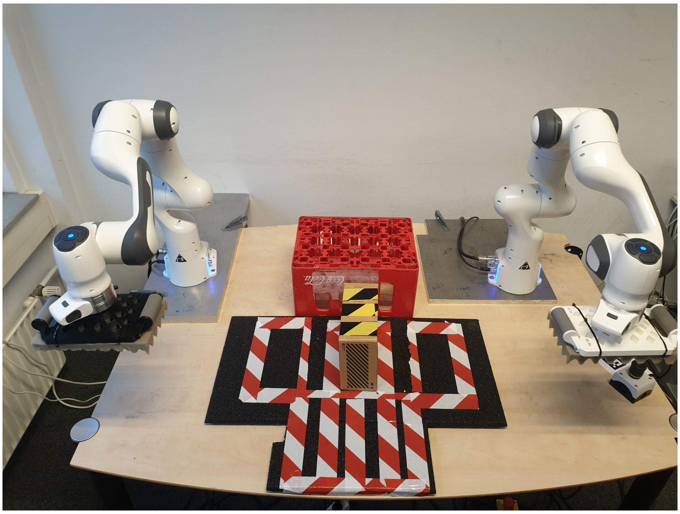


Figure 6.10: Set up used for the user study to compare the performance of non-expert users in performing corrections versus new demonstrations. The workspace is discretized in different regions where the box can be placed or dragged to as a waypoint.

Score	Q1	Q2	Q3	Q4	Q5
0 (strongly disagree)	0	0	0	4	0
1 (disagree)	0	2	2	4	2
2 (slightly disagree)	0	2	4	3	4
3 (slightly agree)	1	3	5	0	1
4 (agree)	2	4	0	1	5
5 (strongly agree)	9	1	1	0	0
mean	4,67	3.00	2,50	1,67	2,75
standard deviation	0,65	1,28	1,09	1,19	1,22

Q1: After giving Kinesthetic demonstration, I feel the robot is performing the task well,
Q2: After providing corrections, I feel the robot is adapting well to the novel situation,
Q3: I feel that giving a bimanual Kinesthetic Demonstration with two arms is tedious for a human teacher,
Q4: Performing Interactive Corrections is LESS physically tiring than giving a completely new demonstration,
Q5: Performing Interactive Corrections is LESS mentally tiring than giving a completely new demonstration

Table 6.1: Likert scale: corrections vs. new demonstrations

as slightly less mentally demanding, probably because they needed to pay attention only to specific segments as opposed to the whole task.

6.6 CONCLUSION

This chapter contributes to the field of bimanual manipulation with an interactive kinesthetic learning framework named SIMPLe. It uses a novel formulation of GP, named GGP, that is computationally efficient and ensures local and global stability of the motion while retaining an estimation of epistemic uncertainties. Thanks to the kernel formulation, the policy encoding can go from purely time-dependent to purely position-dependent or to a combination of both. At the same time, the graph representation of it allows an online update of the time *belief* that, differently from the robot position, cannot be directly measured. The study reports a comparison of a GP with the novel GGP, see Figure 6.3 and an ablation study when the time dependence is considered or not, see Figure 6.5. We conclude that considering the time and properly updating its beliefs allows dealing with more complex and possibly ambiguous demonstrations.

Various technical validation experiments were performed on a real bimanual setup to demonstrate the key functionalities and capabilities of the proposed method. The supplementary user studies gave interesting insights into how humans feel when teaching and correcting a robot with different modalities. This study reported that users are faster and less stressed when performing kinesthetic teaching compared to teleoperation. Furthermore, most users prefer giving corrections to completely new demonstrations.

However, to transfer the learned skills to new situations, i.e. locations of the box to pick, we must develop a way of transporting the learned skill from the demonstration context to the new context. The next chapter will formalize this by exploiting the use of GPs allowing the generalization to of policy for pick and place, dressing, and cleaning tasks while retaining a clear formulation of the uncertainties.

7

Generalization of Task Parameterized Dynamical Systems using Gaussian Process Transportation

Learning from Interactive Demonstrations has revolutionized the way non-expert humans teach robots. It is enough to kinesthetically move the robot around to teach pick-and-place, dressing, or cleaning policies. However, the main challenge is correctly generalizing to novel situations, e.g., different surfaces to clean or different arm postures to dress. This article proposes a novel task parameterization and generalization to transport the original robot policy, i.e., position, velocity, orientation, and stiffness. Unlike the state of the art, only a set of points are tracked during the demonstration and the execution, e.g., a point cloud of the surface to clean. We then propose to fit a non-linear transformation that would deform the space and then the original policy using the paired source and target point sets. The use of function approximators like Gaussian Processes allows us to generalize, or transport, the policy from every space location while estimating the uncertainty of the resulting policy due to the limited points in the task parameterization point set and the reduced number of demonstrations. We compare the algorithm's performance with state-of-the-art task parameterization alternatives and analyze the effect of different function approximators. We also validated the algorithm on robot manipulation tasks, i.e., different posture arm dressing, different location product reshelving, and different shape surface cleaning. A video of the experiments can be found here: `https://youtu.be/FDmWF7K15KU`.

This chapter is based on 🗎 ***Franzese, G.***, *Prakash, R., and Kober, J. (2024). Generalization of Task Parameterized Dynamical Systems using Gaussian Process Transportation*[57].

7.1 Introduction

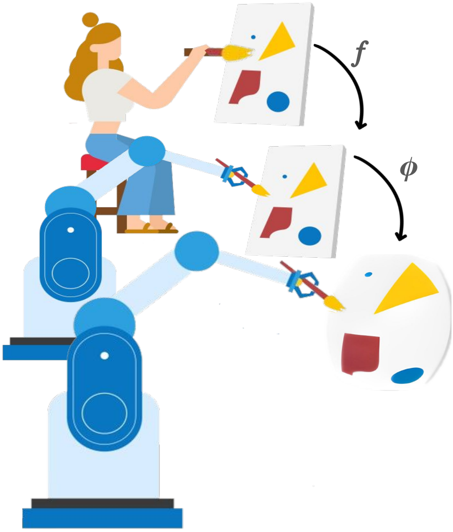


Figure 7.1: Example of Policy Transportation. The human demonstrates to a robot how to perform a task on a flat canvas. Then, the robot, when facing a new curved canvas, “transports” its knowledge in the new situation by adapting the end effector velocity, orientation, and stiffness to correctly adapt the drawing on the new canvas.

One of the main appeals of robot learning from demonstrations is that it enables humans with different levels of robotic expertise to transfer their knowledge and experience about skills and tasks to the robot [38]. This alleviates the need to program such skills by hand, which is tedious, error-prone, and requires an expert. However, one of the long-term challenges of this approach is generalizing the learned behavior to novel situations.

By enhancing the policy with task parameterization [28], robots can generalize their learned knowledge to different variations of the same task, thus promoting scalability and data efficiency in robot learning, allowing robots to learn faster and adapt to new scenarios. For instance, a robot can be trained to clean surfaces with a reduced set of shapes, to dress an arm in a certain configuration, or to pick objects with a certain shape and place them on the right shelf. Ideally, the robot would generalize the learned skill to novel situations without extensive retraining. This chapter proposes a way of transfer, or “transport", the original learned behavior from the old to the new situation.

For example, let us imagine teaching a robot how to draw on a flat canvas, as depicted in Fig. 7.1. The robot can learn to imitate the human on the same flat canvas, however, when the desired surface to reproduce the task is changed, i.e., the blue curvy surface, the user may have to teach the task again on the new surface. Instead, by knowing the correspondence of a set of points from the two surfaces, we propose to learn a function that could generalize the policy from the original parameter distribution to the new one.

The learned function is locally deforming the space according to the new location of the tracked points and this can be used to generalize the learned velocity field, learned stiffness and orientation. Fig. 7.2, depicts how the trajectory and the learned desired dynamics of drawing a letter C depicted as red dots, on a flat surface, depicted in green; the robot would learn the desired behavior, i.e., the velocity depicted as black arrows, form any position of the space. However, when the desired surface to reproduce the task is changed, i.e., the blue curvy surface, the user may have to teach the task again on the new surface. Instead, by knowing the correspondence of a set of points from the two surfaces, we show how to learn a function that could generalize the policy from the original surface shape to the new one.

The proposed algorithm contributes with the formalization and testing of a policy transportation theory that can

- transport the demonstration from the original space to the new space, see Sec. 7.3.2;
- transport the velocity field, end-effector orientations, stiffness and damping by exploiting the derivative of the transportation mapping, see Sec. 7.3.3;
- estimates the final uncertainty due to the reduced set of demonstrations and the estimated uncertainty in the transportation map, see Sec. 7.3.5.

The same algorithm was tested on generalizing complex manipulation tasks like cleaning surfaces with different shapes, picking and placing objects at other locations, and dressing an arm in various configurations, see Sec. 7.5.

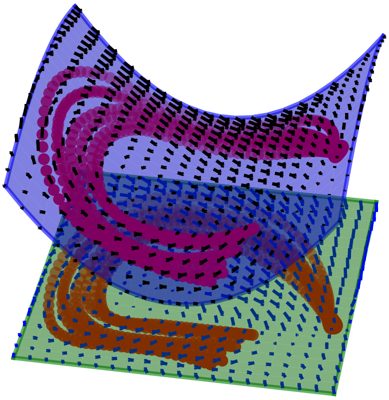


Figure 7.2: The demonstrations (in red) are given on the green surface, and the learned dynamics, depicted as arrows, are learned from them. Later, the demonstration and the dynamics are projected on another curved blue surface using the proposed Gaussian Process Transportation.

7.2 Related Works

One classical method to generalize behavior to new situations involves task parameterization, such as the picked object location, target goal, or via points. This idea of behavior representation and generalization in varying task configurations has been popularly achieved using DMPs [146] through a single or multiple demonstration per task. The DMP model consists of stable second-order linear attractor dynamics with alterable target parameters (end goal or velocities).

An approach to adapt the DMPs via points is addressed in [145], but it demands combining several DMPs for a single task. Alternatively, a roto-translation can be applied to the original dynamical systems according to the tracked frame or points in the environment.

Approaches for modeling and generalizing demonstrations that have shown improved performances with respect to the DMPs are Probabilistic Movement Primitive (ProMP)[125]. ProMPs model the distribution over the demonstrations that capture temporal correlation and correlations between the DoFs using a linear combination of weights and a set of manually designed basis functions. Adaptation to new task parameters or via points is achieved using Gaussian conditioning. While this approach allows modeling the structure and variance of the observed data in the absolute reference frame, the generalization to the new task parameters is satisfactory only within the confidence bound of the demonstration data. For example, showing many demonstrations for different goal points, the probabilistic model can be conditioned on a novel object position and retrieve the most probable trajectory that brings the robot to that final position. However, when learning reactive policies, i.e., a function of the state and not of the phase of the motion, the use of ProMPs is limited since the number of basis functions overgrows with the dimension of the input, limiting its applications.

Kernelized Movement Primitives (KMPs) [77] proposed a non-parametric formulation. This formulation allows modulation of the recorded trajectories to new via points, obtaining the deformation of the original movement primitives given the temporal correlation of the demonstration and the via point, calculated with the kernel function. However, the user must specify the time and the corresponding waypoint to deform the original trajectory or rely on a heuristic that, for example, matches each waypoint with the closest point in the trajectory.

Gaussian Mixture Model (GMM) [30–32] have successfully been employed in modeling demonstration, endowing with a successful generalization in its task parameterized version, i.e. Task-Parameterized Gaussian Mixture Model (TP-GMM) [28]. Given a set of reference frames that are tracked during demonstration and execution, the central idea of TP-GMM is the local projection of the demonstrations in each of the local reference frames and encoding each model as a mixture of Gaussians [28, 34, 76, 149]. The local models are then fused in global coordinates, using the Product of Gaussian (PoG), and a new motion is rolled out from the resulting mixture model. This approach, however, requires many demonstrations to fit the model and does not scale well with the increasing number of task frames. This is because the PoG does not scale well when dealing with many reference frames and can lead to undesirable generalizations.

The generalization of the demonstrations, encoded as a chain of events in a graph, with respect to multiple via points, can be done using Laplacian Editing [105, 123]. It uses the Laplace-Betromi operator, a well-known algorithm in the computer graphics community, to deform meshes [153], to encode geometric trajectory properties and generate deformed trajectories using task constraints, i.e., new via points. The operator ensures a smooth deformation of the trajectory through the via points. However, this approach is very specific for trajectory reshaping and requires explicit knowledge of the new via-point for some trajectory nodes. In this chapter, we relax the need to explicitly specify the new via-points for a specific node in the trajectory since the via-point definition can be error-prone and requires ad-hoc algorithms.

Moreover, while all the previous approaches only address generalizing a demonstration/robot trajectory, the proposed approach relaxes the requirement of generalizing only trajectories. It provides the means also to generalize the velocity field of the original dynamics, as well as the orientation and the stiffness. For generalization, we use a combination of linear transformation and non-linear deformation to transport demonstrations to new situations while estimating the process uncertainty. We validated the algorithm on three use cases: robot reshelving, robot dressing, and robot cleaning.

Robot Reshelving: Authors of [116] propose, within the realm of robotic retail automation, to enable non-expert supermarket employees to teach a robot a reshelving task and then adeptly generalize its learned policy to accommodate diverse task situations. The generalization of the policy for varying object locations is achieved by switching between the dynamical system learned between the object and the goal frame. However, the switching strategy entails having a good prior on when to switch and all the possible implications of generating instability by suddenly changing the policy online. TP-GMM alternatives [28], solved the problem of the switching by obtaining the final GMM as the product of the relative models; however, more than one demonstration is necessary to fit an informative model.

Robot Dressing: Robot dressing is a challenging task since it includes manipulation of deformable objects, and the margin of error to correctly go through the human arm is very low. Task parameterized dynamical system has been applied to learn the dressing task in the robotics research. The dressing demonstrations w.r.t. the wrist and the shoulder of a human arm have been used to learn a dressing policy via DMP [84], HMM [129] and a TP-GMM [181].

Robot Surface Cleaning: Efficient and fast generalization of robotic surface cleaning can be achieved using task-parameterized learning. In [9] the cleaning dynamics is the sum of two dynamical systems, one that learns the desired motion on the surface and another that computes the modulation term on the desired force to apply on the perpendicular direction of the surface (where the shape is known a priori). This second term is learned as a non-linear function that allows learning larger forces in a region of the surface compared to others. The shape of the surface can also be estimated using the wrench measured with a force-torque sensor attached at the end-effector; for instance, [13] generalizes the polishing task on the novel curved surface by adapting the orientation and the direction of the contact force such that to minimize perceived torque.

The following section will provide some background on the main concepts necessary for learning a policy from demonstration and generalize it using Gaussian Process Transportation.

7.3 Policy Transportation

We learn and correct a manipulation policy from an interactive demonstration [38], using, for example, a GP as the function approximator to fit the demonstration dynamics.

However, although the task execution would succeed from any given starting configuration of the robot, it will fail to generalize if the task is changed, e.g., if the object to pick is moved or if the robot faces a differently shaped surface to clean.

Intending to find a task parameterization that scales from pick-and-place to continuous surface, we assume to track a set of environment-specific points that are descriptive of the situation. For example, when picking a box, the eight corners are tracked, or when cleaning a surface, a point cloud representation is used. This is the most straightforward yet most general task parameterization while being technically feasible, given the current development of LiDAR and depth camera technology.

We define the tracked N point recorded in the demonstration scenario as the source distribution, i.e.,

$$\boldsymbol{\mathcal{S}} = \{(x_{s,i}, y_{s,i}, z_{s,i})\}_{i=1}^{N}$$

while the moved points in the new scenario are defined as the target distribution, i.e.,

$$\boldsymbol{\mathcal{T}} = \{(x_{t,i}, y_{t,i}, z_{t,i})\}_{i=1}^{N}.$$

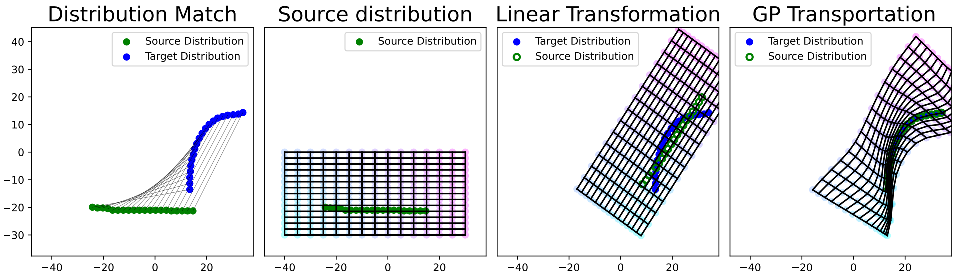


Figure 7.3: 2D transportation. **Distribution match** depicts the source and the target distribution correspondence used to train the transportation function. **Source Distribution** depicts a grid of points in the original space. **Linear Transformation** shows the effect on the original grid when only a linear transformation is used to match source and target distribution. **GP Transportation** captures the deformation of the space when the source points are forced to match the target ones.

We assume that the points of target and source distribution are already paired. Many algorithms are available to (optimally) pair the two distributions [42]; hence this is not the focus of this work.

We define a map $\boldsymbol{\phi}$ such that each point s_i in S is paired with one and only one point t_j in T. This can be represented as:

$$\boldsymbol{\phi} : \boldsymbol{\mathcal{S}} \to \boldsymbol{\mathcal{T}}$$

$$s_i \mapsto t_j$$

where $i, j \in \{1, 2, \ldots, n\}$. We aim to find the function that maps from the source space to the target space, given the evidence of the input-output pairs from the source and target distribution. Estimating a continuous process allows the deformation of the complete space to match the source and the target distribution, as depicted in Fig. 7.3.

The structure of the function $\boldsymbol{\phi}$ that we want to approximate can be any nonlinear function that maps any point of the Cartesian space to itself, e.g., $\boldsymbol{\phi} : \mathbb{R}^3 \to \mathbb{R}^3$. However, in the context of this article, we consider the transportation function to have the following definition,

$$\boldsymbol{\phi}(\boldsymbol{x}) := \boldsymbol{\lambda}(\boldsymbol{x}) + \boldsymbol{\psi}(\boldsymbol{\lambda}(\boldsymbol{x})) \tag{7.1}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$ are, respectively, a linear and nonlinear transformation. The fitting of the function is made in two steps: first the linear transformation $\boldsymbol{\lambda}(\boldsymbol{x})$ is obtained, and then the nonlinear transformation $\boldsymbol{\psi}(\boldsymbol{\lambda}(\boldsymbol{x}))$ is fitted on the residual error.

7.3.1 Linear Transformation

To fit the optimal rotation matrix between the source and the target distribution, the centered source and target distribution are used as labels for the fitting of the function $\boldsymbol{\lambda}$, i.e.

$$\underbrace{\mathcal{T}-\bar{\mathcal{T}}}_{y_{\text{label}}} = \lambda(\underbrace{\mathcal{S}-\bar{\mathcal{S}}}_{x_{\text{label}}})$$

where $\bar{\mathcal{S}}$ and $\bar{\mathcal{T}}$ are the centroid of the source and the target distribution, respectively. We can find the rotation between the two centered distributions using the Singular Value Decomposition (SVD) imposing

$$\boldsymbol{U}\Sigma\boldsymbol{V}^\top = (\mathcal{S}-\bar{\mathcal{S}})^\top(\mathcal{T}-\bar{\mathcal{T}})$$

and the rotation matrix is defined as,

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{U}^\top,$$

however, if $\det(\boldsymbol{P}) < 0$, the last column of $\boldsymbol{V}$ is flipped in sign, and the computation of the rotation matrix is repeated. This ensures the transformation is a proper rotation matrix without any reflection; see [14] for more details. Hence, the linear transformation on any point in the space can be computed as

$$\lambda(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}-\bar{\mathcal{S}}) + \bar{\mathcal{T}}. \tag{7.2}$$

Fig. 7.3c shows a linear transformation of the source and a grid of points from the original space depicted in Fig. 7.3b.

7

7.3.2 Non-linear Transportation

After fitting the linear transformation of Eq. (7.2), the residual transformation is obtained by substituting the source, target points, and the fitted linear function in Eq. (7.1), obtaining that

$$\underbrace{\mathcal{T}-\lambda(\mathcal{S})}_{y_{\text{label}}} = \psi(\underbrace{\lambda(\mathcal{S})}_{x_{\text{label}}}).$$

The nonlinear function $\boldsymbol{\psi}$ can be any nonlinear regressor, such as a Neural Network, a Random Forest, a Gaussian Process, etc. However, the inducting bias given by the nature of the nonlinear function will affect the regression output when going out of distribution, i.e., far away from the given data. For example, suppose the function is approximate with a GP with a distance-based kernel k, such as a square exponential kernel.

If the prior distribution is set to be a zero-mean function when making predictions in regions of the space far away from the source distribution points, the final transportation converges to just being a linear transformation, see Fig. 7.3d. Knowing the out-of-distribution (o.o.d.) properties of the transportation policy is desirable, considering that we will transport points that are not necessarily close to the point of the source/target distribution.

$$\begin{tikzcd}
x \arrow[r, "f"] \arrow[d, "\phi"'] & \dot{x} \arrow[d, "\frac{\partial \phi}{\partial x}"] \\
\hat{x} \arrow[r, "\hat{f}"'] & \dot{\hat{x}}
\end{tikzcd}$$

Figure 7.4: Mathematical Scheme of Policy Transportation.

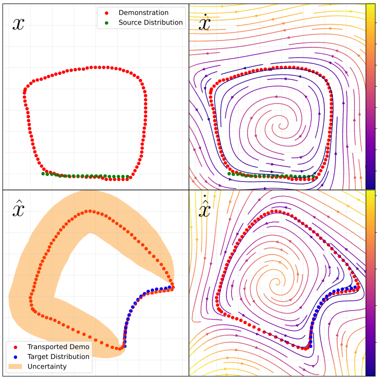
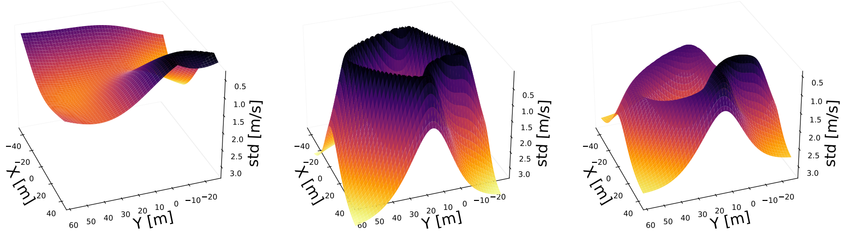


Figure 7.5: Graphical representation of the mathematical scheme of policy transportation.



Transportation Uncertainty

Epistemic Uncertainty

Total Uncertainty

Figure 7.6: Standard Deviation quantification on the velocity field. **Transportation Uncertainty** was computed with Eq. 3.3 and quantifies the (heteroscedastic) uncertainty on the transported label (velocity) corresponding to the transported demonstration. **Epistemic Uncertainty** is the resulting model uncertainty when fitting the new policy $\hat{f}$. **Total Uncertainty** is the resulting standard deviation after computing the variance sum of transportation and epistemic uncertainties.

7.3.3 Transportation of the Dynamics

Although the transportation map allows the transport of any point of the original demonstration in the new situation, for example, to generalize the demo on cleaning a new surface, we still have not formulated a transportation function for the velocity field $\dot{\boldsymbol{x}}$. It is not as trivial as computing the numerical differentiation of the transported trajectories. We consider the policy labels as independent points, no longer part of a trajectory. This allows us to learn from multiple demonstrations and to change the velocity label connected to them if providing (teleoperated) feedback or aggregating new data from interactive demonstrations [55]. Nevertheless, the partial derivative of the transportation mapping can be exploited in the velocity field generalization. Given the transportation function defined in the source space and projecting in the target space, i.e.,

$$\hat{\boldsymbol{x}} = \boldsymbol{\phi}(\boldsymbol{x})$$

by differentiating w.r.t. time on both sides and using the chain rule, we obtain the velocity field in the transported space as

$$\dot{\hat{\boldsymbol{x}}} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x})\dot{\boldsymbol{x}}$$

where the Jacobian matrix, using the definition of Eq. (7.1), can be defined as

$$\boldsymbol{J}(\boldsymbol{x}) := \frac{\partial \boldsymbol{\lambda}(\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{\psi}(\boldsymbol{x})}{\partial \boldsymbol{\lambda}(\boldsymbol{x})} \frac{\partial \boldsymbol{\lambda}(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

where $\frac{\partial \boldsymbol{\lambda}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{P}$ and $\frac{\partial \boldsymbol{\psi}(\boldsymbol{x})}{\partial \boldsymbol{\lambda}(\boldsymbol{x})}$ can be obtained using automatic differentiation of the chosen regressor. In the following sections, we will simplify notation by omitting the explicit dependence of $\boldsymbol{J}$ on $\boldsymbol{x}$.

7.3.4 Robot Orientation and Stiffness Generalization

However, when learning and controlling the Cartesian robot pose, we must also generalize the desired end-effector orientation.

Let us consider the end effector to be a vector of infinitesimal length with the base $\boldsymbol{x}_0$ on the end effector position and pointing in the direction of the robot orientation during the demonstration, $\boldsymbol{R}_{ee}$. The transportation of the tip of the vector can be obtained using the Taylor approximation of Eq. (7.1), according to

$$\hat{\boldsymbol{x}}_{\text{tip}} = \boldsymbol{\phi}(\boldsymbol{x}_{\text{tip}}) \approx \boldsymbol{\phi}(\boldsymbol{x}_0) + \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{\epsilon} = \boldsymbol{\phi}(\boldsymbol{x}_0) + \boldsymbol{J}\boldsymbol{R}_{ee}\boldsymbol{\epsilon}_0 \tag{7.3}$$

where $\boldsymbol{R}_{ee} \in \mathbb{R}^{3\times 3}$ is the original robot orientation, $\boldsymbol{\epsilon}_0$ is a vector with an infinitesimal dimension that has zero orientation. From Eq. (7.3), it is readily apparent that the transported orientation matrix of the robot end-effector becomes

$$\hat{\boldsymbol{R}}_{ee} := \boldsymbol{J}\boldsymbol{R}_{ee}.$$

The transported orientation matrix needs to be orthogonal with the determinant equal to 1; hence, the pre-multiplication matrix $\boldsymbol{J}$ must have the same properties. We enforce this by normalizing $\boldsymbol{J}$ with its determinant and finding the corresponding orthogonal matrix with a QR decomposition.

Additionally, when implementing policies on a Cartesian impedance control, the stiffness $\boldsymbol{\mathcal{K}}$ and the damping matrix $\boldsymbol{\mathcal{D}}$ must also be transported. The change of coordinates of the stiffness and the damping follows from the transportation of the robot-applied force on the environment found using Hooke's law, i.e.,

$$\hat{\boldsymbol{F}} = \hat{\boldsymbol{\mathcal{K}}}\Delta\hat{\boldsymbol{x}} = \hat{\boldsymbol{\mathcal{K}}}\overbrace{\boldsymbol{J}\Delta\boldsymbol{x}}^{\Delta\hat{x}} = \boldsymbol{J}\overbrace{\boldsymbol{\mathcal{K}}\Delta\boldsymbol{x}}^{\boldsymbol{F}}.$$

Hence, the generalization of the stiffness matrix becomes

$$\hat{\boldsymbol{\mathcal{K}}} = \boldsymbol{J}\boldsymbol{\mathcal{K}}\boldsymbol{J}^T$$

and following a similar reasoning for the damping matrix, we obtain,

$$\hat{\boldsymbol{\mathcal{D}}} = \boldsymbol{J}\boldsymbol{\mathcal{D}}\boldsymbol{J}^T,$$

considering that the inverse of an orthogonal matrix is equal to the transpose of the matrix itself.

7.3.5 TRANSPORTATION UNCERTAINTY

A probabilistic function approximator, like a GP, will also provide the uncertainty on transportation output that can be propagated in the transported dynamical system.

In particular, a GP derivative is also a GP [172] and its existence will depend on the differentiability of the kernel function. The correlation between derivative samples can be expressed as the second partial derivative $k^{11} = \frac{\partial^2}{\partial x_i \partial x_j} k(x_i, x_j)$ while the correlation between derivative samples and function samples is $k^{10} = \frac{\partial}{\partial x_i} k(x_i, x_j)$. Thus, the mean and variance prediction of the derivative of the Gaussian Process become

$$\begin{aligned} \boldsymbol{\mu}' &= \boldsymbol{K}^{10}_{X_*,X}(\boldsymbol{K}_{X,X} + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{y} \\ \Sigma' &= \boldsymbol{K}^{11}_{X_*,X_*} - \boldsymbol{K}^{10}_{X_*,X}(\boldsymbol{K}_{X,X} + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{K}^{01}_{X,X_*}. \end{aligned} \tag{7.4}$$

The uncertainty quantification of the transportation policy becomes essential for calculating the final uncertainty on the control variable, e.g., the velocity. In Fig. 7.5, the uncertainty is also displayed as a shaded area around the demonstration and as the “warmness" of the color in the vector field.

The uncertainty of the velocity labels is due to the propagation of the original labels through the derivative of the (uncertain) transportation map, see Fig. 7.5, i.e.,

$$\Sigma_{\hat{\boldsymbol{x}}} = \Sigma_{\frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial \boldsymbol{x}}} \dot{\boldsymbol{x}}^2.$$

given the definition of the weighted sum of Gaussian variables [46].

Hence, considering that the labels are uncertain, the prediction of the resulting heteroscedastic GP [103] can be computed as the sum of the epistemic and (variable) aleatoric uncertainty, that is

$$\Sigma_{\dot{\hat{x}}} = \Sigma_{\hat{f}} + \Sigma_{\hat{\boldsymbol{x}}}.$$

Fig. 7.6 depicts the *transportation uncertainty* on the norm of the velocity, calculated with Eq. (7.4) and the epistemic uncertainty of the model $\hat{f}$, computed with Eq. (3.3) using transported position and transported velocities labels. From Fig. 7.5 and 7.6, it is possible to appreciate that the transportation uncertainties grow when evaluating in regions that are far away from the task parameterization points since the transportation is less certain when going far away from the distribution data; on the other hand, the epistemic uncertainty grows when evaluating in points that are far from the transported demonstration.

In conclusion, the sum of the two uncertainty fields in Fig. 7.6 grows either when we go far away from the (transported) demonstration or away from the points of the source/target distribution.

7.4 2-D SIMULATIONS AND COMPARISONS

The availability of (calibrated) uncertainties is an important feature to improve the trustworthiness in deploying robot motion generalization. In this section, we evaluate Gaussian Process Transportation (GPT) on generalizing the demonstration in a 2D surface cleaning task and on a reference frame-to-frame motion generalization. The goal for these simulated experiments is

- to illustrate and compare how the generalization process differs when employing regressors other than a Gaussian Process or methodologies from the state-of-the-art while generalizing a cyclic demonstration that approaches and then retreats from the surface “to clean", in Sec. 7.4.1;
- assess and compare GPT ’s ability to generalize in multi-reference frame tasks, measuring its performance against state-of-the-art algorithms, in Sec. 7.4.2.

7.4.1 2-D SURFACE CLEANING

Fig. 7.5 visualizes the transportation of the given demonstration, in red, from the source to the target space, using the transportation map $\boldsymbol{\phi}$ where the non-linear component was chosen to be a GP, given the out-of-distribution prediction and the calibrated uncertainty quantification. However, other state-of-the-art function approximators can be used to fit the transportation function without loss of generality. To ensure a fair comparison, the mean linear transformation, i.e., $\boldsymbol{\gamma}$, is applied to all trajectories before using the different methods to perform the non-linear transportation. Table 7.1 summarizes the method with their properties, while Fig. 7.7 shows the generalization of the demonstration when these methods are used.

Kernelized Movement Primitives (KMP) [77], in this study, fits the motion as a function of time while Laplacian editing (LE) [123] considers the topology of the demonstration to be a chain, i.e., a graph where only consecutive vertices are connected with an edge or as a ring, when the demonstration is periodic, i.e. also starting and ending nodes are connected, like in Fig. 7.7. Hence, every point of the source distribution is matched with the closest point of the demonstration. Then, each point of the trajectory, or the graph, is moved, knowing the new desired target location of the matched points. Hence, LE and KMP do not provide any uncertainty on the transportation process.

All the other transportation regressors, a part of the GP, are ensembles of popular regression functions, i.e., Ensemble Random Forest (E-RF), Ensemble Neural Networks (E-NN), and Ensemble Neural Flows (E-NF). An ensemble is a collection of multiple individual models, trained independently, whose combined predictions are used to estimate a distribution on the prediction, i.e., mean and variance. In this example, Neural Networks are simple multi-layer perceptions while Neural Flows [93] are bijective neural networks, i.e., flows, usually used to learn a mapping from a simple probability distribution to a more complex target distribution. Fig. 7.7 depicts the mean and the uncertainty bounds of 2-σ for the transported trajectories when using ensembles and GPs. The bounds are computed from the different fitted models in the ensembles while it is computed analytically for the GP, and the depicted GP samples of Fig. 7.7 are drawn from the posterior distribution.

From Fig. 7.7, the reader can appreciate how the GP is the only regressor with well-calibrated and unbiased epistemic uncertainty quantification and minimal mean prediction

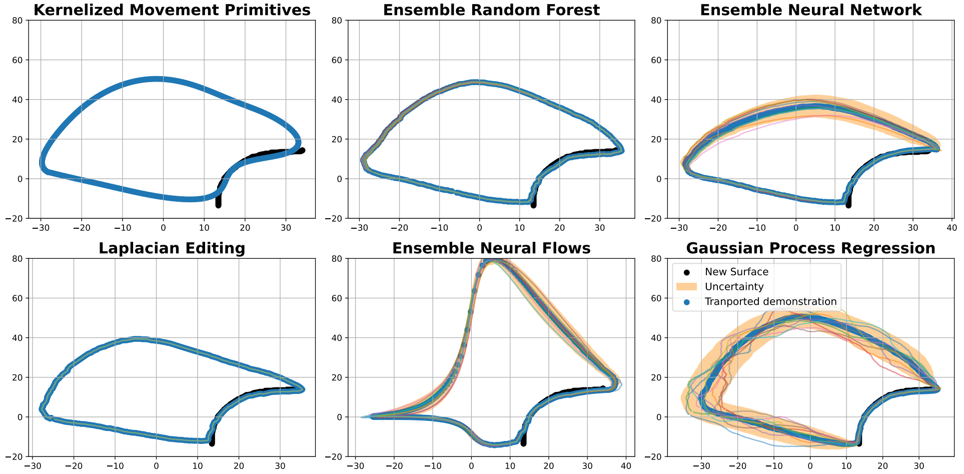


Figure 7.7: Qualitative comparison of transportation of demonstration in target space for 2D surface cleaning. The colored lines are the samples of the final transported trajectory policy, i.e. $\hat{x} = \phi(x)$, and the orange area is 2 standard deviation. The black curve is the 1-D surface to clean.

Table 7.1: Summary table of different methods used to transportation of trajectories to different surfaces.

Method	Modality	Vel. Gen.	Transportation Uncertainty
KMP [77]	way-points	✗	✗
LE [123]	way-points	✗	✗
E-RF [25]	continuous	✓	estimated
E-NF [135]	continuous	✓	estimated
E-NN [100]	continuous	✓	estimated
GP [172]	continuous	✓	analytical

distortion of the trajectory when transporting points far away from the source distribution. For example, the E-NN, has higher uncertainty on the right side of the demonstration, even though the points are at the same distance from the surface, while E-RF generates an undistorted overconfident transformation, i.e., the uncertainty does not grow when going out of distribution.

7.4.2 Multiple Reference Frames

In the literature, one of the main applications of task parameterization is the generalization w.r.t. one or more reference frames. For example, if we teach a robot how to pour water into a glass, we want the robot to automatically generalize the motion w.r.t. any glass position. The task, in this particular case, can be parameterized with the location of the reference frames of each object, which is necessary to track for a successful generalization of the motion. Typically, the motion is projected in any of the reference frames, and a policy is learned w.r.t. each of the frames, leaving out the decision on the relevance of each frame for every timestep. Task-Parametrized Gaussian Mixture Model (TP-GMM) learns a GMM model for the projected demonstration for each of the frames and, during executions, the Gaussians of each frame are combined using the property that the Product of Gaussians (PoG) is still a Gaussian, see [28] for more details. Given the GMM, different control formulations are possible, for example only relying on the current state of the system, i.e., $\Delta \boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{x}_i)$ [8] [28] or by using a HMM formulation that also considers the progress during the execution of the trajectory $\boldsymbol{x}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\alpha}_i)$, where $\boldsymbol{\alpha_i}$, in the context of a mixture mode, selects the properties of the model (mean and variance) that can be used in a tracking algorithm, such as a Linear Quadratic Regulator (LQR) [28, 33]. However, the latent transition matrix between the different states of the HMM is unknown. They need to be estimated using a forward pass algorithm, i.e., the Viterbi algorithm [33], that requires an initial guess trajectory to infer the most likely state transition that generated that initial guess and again generate the most likely motion according to the model. However, having an initial guess can be prohibitive when evaluating the movement in a novel configuration of starting and goal frames.

Differently from these task-parameterized approaches, the proposed method does not track only the reference frame but a set of points that are relevant to the starting and goal object. To guarantee a fair comparison with the state of the art, in Fig. 7.8, when generalizing using GPT, only 5 points are tracked w.r.t. each reference frame, capturing the position but also the local orientation of the frames. In Fig. 7.8, what we describe as DMP uses the same mathematical structure of GPT but only relies on a linear transformation, which is why the result is not able to capture the non-linear deformation due to the frame orientation. One of the main perks of the proposed method is the ability to generalize any dynamical system generated by even only one demonstration, unlike GMM-based methods where, to capture a meaningful mixture model, at least two diverse demonstrations need to be provided. Additionally, the GMM uncertainty of the final multi-frame model that results from the PoG does capture the uncertainty of the transportation, while, as depicted in Fig. 7.8, the GP transportation results in growing uncertainties when transporting points of the demonstrations that are less correlated with the source-target points.

Fig. 7.8 highlights the discrepancy in the performance of GMM methods on the training set and the test set. At the same time, the reproduction of a known combination of the frames results in accurate rollouts of the policies both when executing them as a dynamical system (TP-GMM)[1] [8] then as an optimal tracking problem of a multi-transition Hidden Markov Model (HMM)[2] [33], when evaluating on the test set, generated on random reorganization of the frames, the resulting trajectories do not successfully reach the goal frame neither in position or orientation. To quantify and compare the different methods, we conducted a quantitative analysis comparing the generalization on known trajectories from the demonstration set or the reaching performance on a randomly generated frame set.

Quantitative Analysis Fig. 7.9 shows the box plot that compares the performance of the different models, i.e., TP-GMM, HMM, DMP , and GPT , on the training set. Nine demonstrations are available for different configurations of the starting and goal frame. When training the GMM models, i.e., TP-GMM and HMM, a subset of demonstration m is randomly chosen from the training set and compared with the remaining $(9-m)$ demonstrations when evaluating the model in that unknown situation; the number of used

[1] `https://github.com/BatyaGG/Task-Parameterized-Gaussian-Mixture-Model`

[2] `https://gitlab.idiap.ch/rli/pbdlib-python/-/blob/master/notebooks/`

demonstration is highlighted as an apex, e.g., HMM_6 means that we used an HMM model with six demonstrations. When training the transportation models, linear or non-linear, i.e., DMP or GPT , only one demonstration is randomly picked from the training set and compared with the other eight unseen situations. For each model, the random selection of demonstration and comparison is repeated 20 times. Five metrics are used to compare the rollout trajectory and the actual demonstration:

- Frechet distance that does not consider any knowledge of time but finds the maximum distance among all the possible closest pairs among the two curves [83];
- area between the two curves that constructs quadrilaterals between two curves and calculates the area for each quadrilateral [83];
- Dynamic Time Warping (DTW) that computes the cumulative distance between all points in the trajectory [83];
- final position error, computed as the Euclidean distance between the final point of the trajectory and the rollout;
- final trajectory angle, that computes the approach “docking” angle of the trajectory. A low error in the angle distance means that the reproduced trajectory approaches the goal from the same direction as the provided demonstration in the same circumstance.

Considering that we have many models that can behave differently according to the amount and quality of the demonstration, it is not straightforward to deduce any conclusions on which method is statistically better from the boxplot of Fig. 7.9. For this reason, we run a U test, also known as the Mann-Whitney non-parametric test [111], to deduce if the distribution of results of each of the methods is statistically lower ($p < 0.05$) than each of the others. When computing the U test between two methods, in case of a statistical difference, the winning method gets one point. The numbers on top of the figure for each of the methods indicate the performance ranking, i.e., the method that obtained the most points when computing the U-test is going to be first in the ranking. When more methods share the same position in the ranking, it simply means they were significantly better than the same amount of other methods during the comparisons.

Fig. 7.9 shows that for Frechet, final position and orientation error, GPT (trained with a single demonstration) performs the best. In contrast, for Area btw the curves and DTW, GPT performs equally or better than GMM and HMM models trained with five demonstrations.

Finally, Fig. 7.10 shows the box plot and ranking for the model evaluated in a test set with randomly placed frames, and GPT performs statistically better than any other method when reaching the right goal and from the right direction.

7.4.3 Multi-source single-target generalization

Fig. 7.8, in the column of Gaussian Process Transportation, depicts the generalization of a single demonstration from one single 2-frame source to multiple 2-frame targets. Although, in Fig. 7.2, we already depicted the generalization of many demonstrations and the learned dynamics from one source surface to another target, we still did not mention the generalization from multi-source to a single target. When dealing with n source distributions,

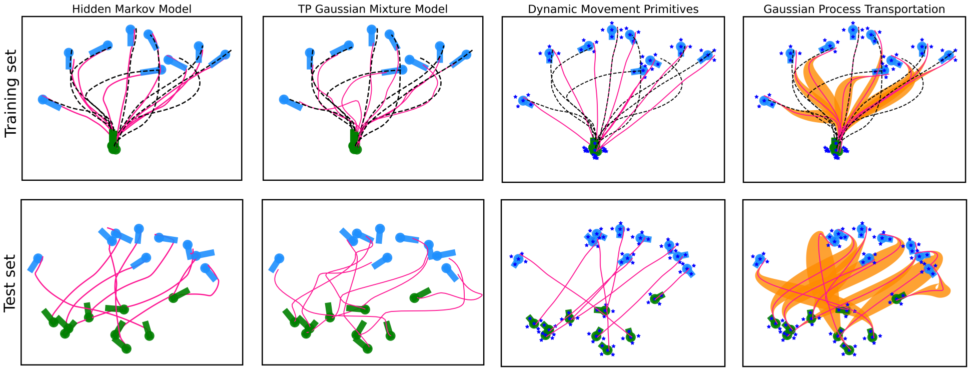


Figure 7.8: Qualitative comparison of multi-reference frame parameterization. Comparison between HMM, TP-GMM, DMP s, and the proposed method. The first raw compared the performance in the reproducing of the training set demonstration, depicted as a dashed black line, where both HMM and TP-GMM are training using all the nine demonstrations while DMP and GPT are only trained with one demonstration (the central one) and generalized for each of the frames. In the second row, a random perturbation is applied to each frame, and the model is queried on the most likely trajectory. For GPT, the uncertainty in the generalization is depicted with the orange areas. The blue stars are the points tracked during the motion, given that the proposed method does not rely on reference frames but only on source and target points.

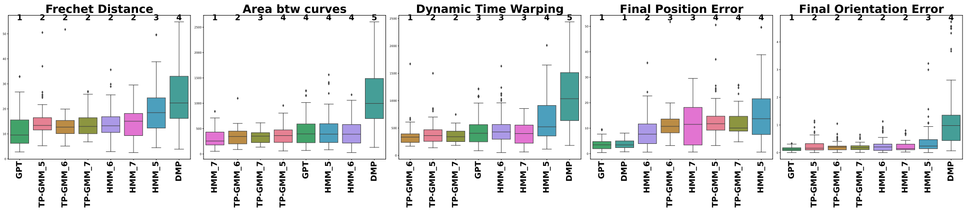


Figure 7.9: Box plot results and performance ranking on frame configuration from the training set. The number on top of each box plot is the position of the method in the performance ranking.

we fit n different transportation functions ϕ, each trained with different source points but the same target points.

Fig. 7.11 highlights how the many demonstrations given in different frame configurations can be transported in the same target frame and how we can extract a reactive policy, encoded in the vector field, as a function of the global position [55].

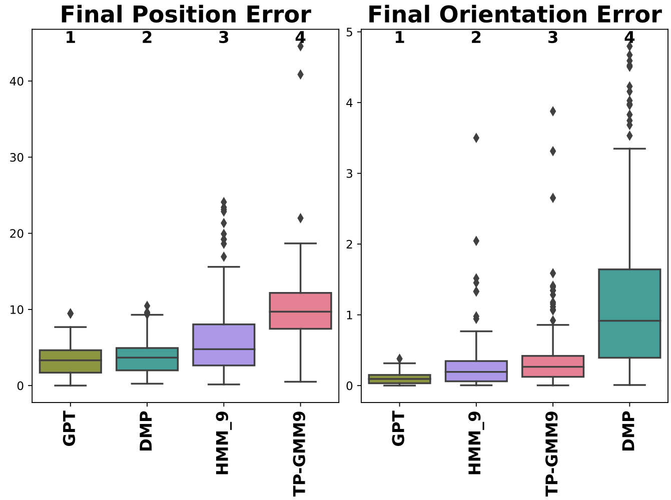


Figure 7.10: Box plot results and performance ranking on randomly generated frame configurations.

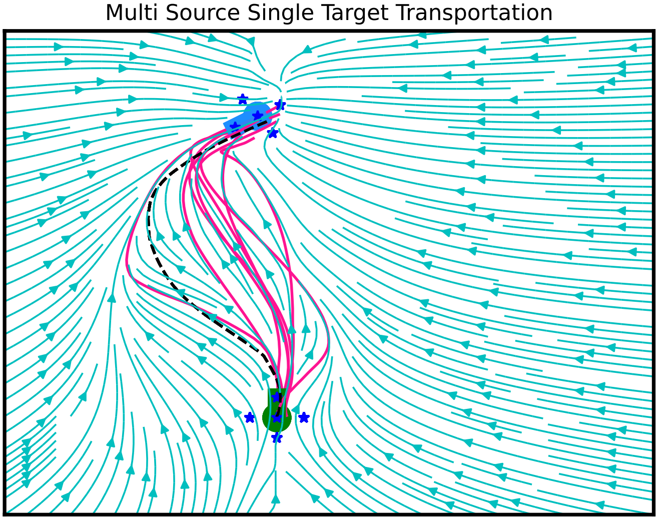


Figure 7.11: Multi-source single target generalization. Demonstrations from different frame positions (see Fig. 7.8 are transported on a single target multi-frame configuration (unknown from the training set). The dashed line is the given human demonstration in that configuration. The vector field is the resulting dynamics learned also with a Gaussian Process with minimization of uncertainties from [55].

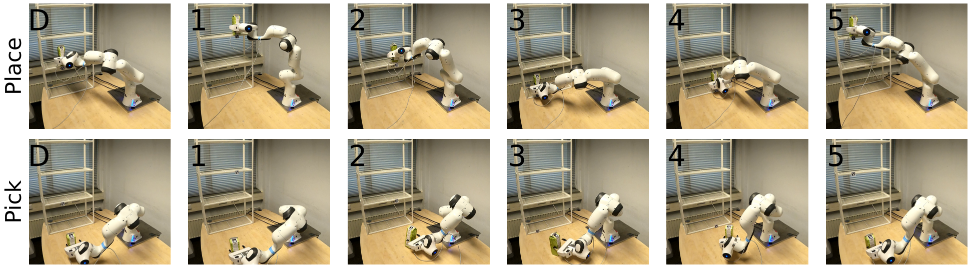


Figure 7.12: Generalization of the reshelving task. The first column is the robot reproduction in the demonstration scenario.

7.5 Real Robot Validation

To validate the proposed method on real manipulation tasks, we selected three challenging tasks, i.e., robot reshelving (Sec. 7.5.1), dressing (Sec. 7.5.2) and cleaning (Sec. 7.5.3), to teach as a single demonstration and generalize it in different scenarios. These are all tasks where the training set will never be similar to the test set; for example, when dressing a human, the configuration and shape of the arm may change, and we expect the robot to generalize the behavior accordingly. We controlled a Franka Robot using a Cartesian impedance control[3].

The dynamics of the demonstrations are learned with a non-parametric function approximation for motor learning that uses a joint position-time encoding, proposed in [56]. The distance from the next attractor position and orientation are a function of the current robot position. Our goal is to show how the proposed transportation theory can correctly generalize the pose, velocity, and stiffness of the robot. The following sections will summarize the robot validation experiments. A video of all the experiments can be found at `https://youtu.be/FDmWF7K15KU`.

7.5.1 Robot Reshelving

Robot reshelving refers to picking an object in one location, moving it, and placing it in a desired position on a shelf.

Our assumptions for the problem are:

- one global frame dynamical system is learned from a single demonstration and transported in the different object/goal configurations;
- corner points of the objects and the shelf slot are tracked rather than position/orientation.

Fig. 7.12 depicts the experimental setup where a milk box, with an AprilTag [170] on it, has to be positioned on a compartment on a shelf, also marked by another tag. Before the demonstrations or execution, the robot searches for any frames in the spaces using the camera attached to its end-effector. For every frame, the transportation policy extracts a cube's center and corners with predefined side dimensions as the markers. Fig. 7.12 shows how the

[3] `https://github.com/franzesegiovanni/franka_human_friendly_controllers`

Frame	x [m]	y [m]	z [m]	yaw [deg]
Object	0.225	0.366	-	94.6
Goal	0.337	0.036	0.675	-

Table 7.2: Range of Variability for Object and Goal Frames.

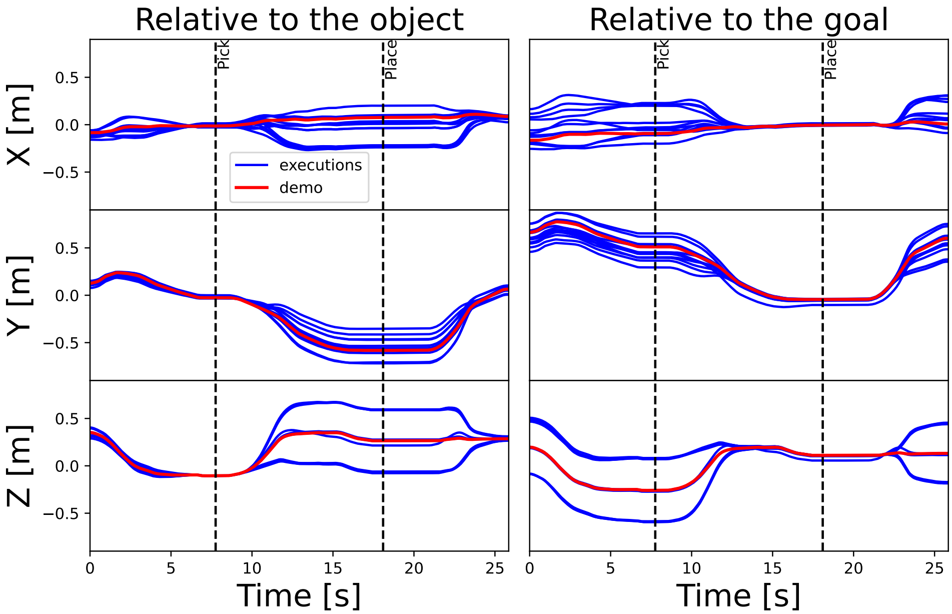


Figure 7.13: Relative position of the end-effector w.r.t. the initial object and the goal position during multiple generalizations rollouts in robot reshelving.

demonstration for reshelving on the left of the central compartment can be generalized to any other floor, both on the left and right. We randomized the object position and orientation and the goal on the shelf ten different times, all successfully generalized. Table 7.2 shows the range of x,y,z, and yaw angles of the object and goal markers during the ten different executions, while Fig. 7.13 depicts the relative position w.r.t. the object and the frame of the different rollouts; from the figure it is possible to appreciate how the execution lines converge on the (initial) object position when picking and on the goal position when placing the object.

7.5.2 Robot Dressing

The task of dressing is a primary task in elderly care. It consists of pulling a deformable sleeve over the posture of a human arm. Complicated motions need to be executed by the robot to increase the dressing success rate, i.e., reaching the shoulder without getting stuck or exercising too large force on the arm. Fig. 7.14 depicts the robot experimental setup where an articulated mannequin is posed in different shoulder positions and arm configurations. Four AprilTags [170] are glued on the arm, shoulder, elbow, wrist, and hand, captured by

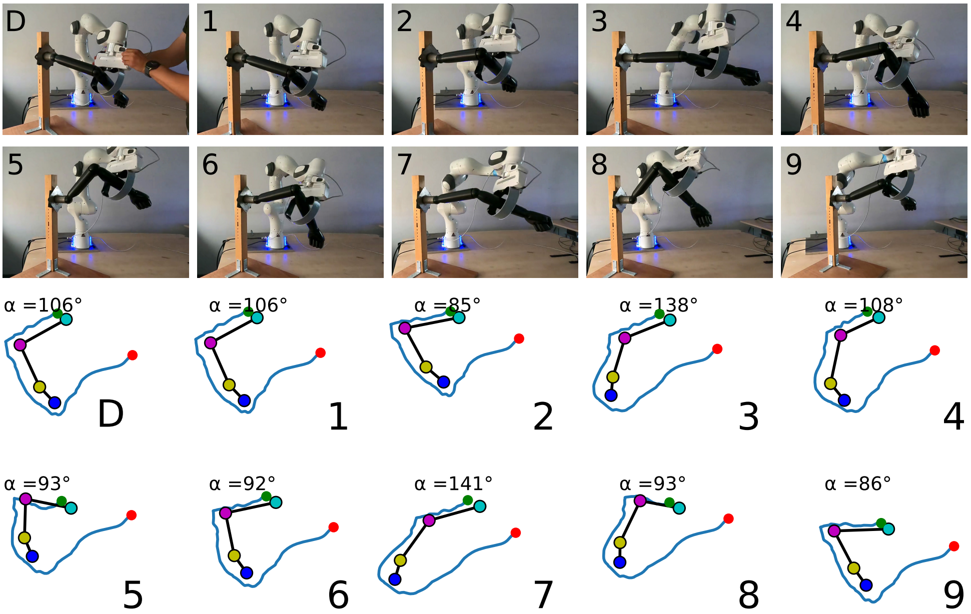


Figure 7.14: Dressing policy generalization. Cyan marker is the shoulder, magenta is the elbow, yellow is the wrist, and blue is the hand. The blue rollout end-effector trajectory starts from the red dot and finishes with the green dot. α is the angle of the elbow; the smaller the angles, the more complicated the generalization would be.

the camera on the robot wrist at the beginning of each demonstration/execution. From the markers, only the position is extracted this time. The piece of cloth is pinched in the end effector by the user before starting the experiments. We leverage the assumption that the pose will not change during the demonstration; however, it is worth mentioning that the arm structure is not fixed on the table, so if the generalization is not good and the robot maliciously touches the arm, the resulting displacement would result in unsuccessful dressing. Only one demonstration was given to the robot. Then, the arm was reset for a different range of x,y positions of the shoulder and configuration of the arm. The ranges of variation of the task parameters are $\Delta x_{\text{shoulder}} = 0.122$ [m], $\Delta y_{\text{shoulder}} = 0.259$ [m], $\Delta\alpha = 56$ [deg], where α is the angle that the elbow intercepts with the connecting line between the shoulder and the wrist. A fully stretched arm (easy pose to dress) has $\alpha = 180$, and when the hand touches the shoulder (impossible pose to dress) $\alpha = 0$. The policy transportation was able to generalize the policy for every requested arm configuration.

7.5.3 Robot Surface Cleaning

Surface cleaning/grinding tasks require robots to not only track surface shapes but also apply the right amount of force for successful cleaning/grinding. Robotic cleaning or grinding involves automated machines equipped with specialized tools to perform cleaning tasks.

In this experiment, we want to show that

- we can learn a general policy that may involve polishing phases and free movement phases;

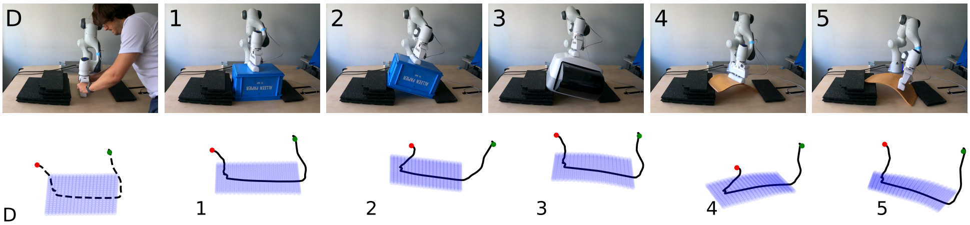


Figure 7.15: Point cloud, demonstration and rollouts of the generalized motion in cleaning tasks.

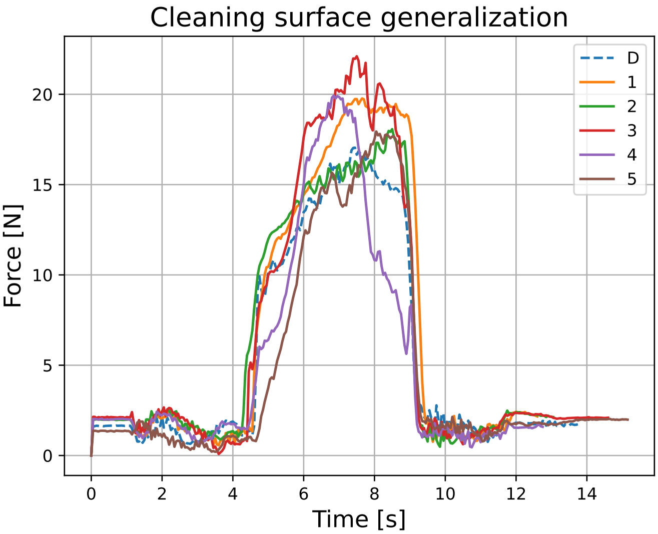


Figure 7.16: Norm of the force perceived [N] from the end-effector when executing the transported dynamics on the new surfaces.

- we do not need any force sensors to align to the surface;
- the surface is unknown, and only a point cloud is obtained from the camera sensors.

One of the main advantages of the proposed method is that it does not need to reconstruct the surface but only learns the map from the source to the target pointcloud. The deformation between the source and the target surface point cloud is modeled using a Sparse Variational Gaussian Process Transportation (SV-GPT) to generalize the demonstrated policy position, orientation, and stiffness profile for a successful cleaning task. Given the large number of points in the source and target point cloud, i.e., 400 points, using a reduced set of inducing points, i.e., 100, makes fitting the transportation model more computationally efficient.

Fig. 7.15 depicts the teaching of a cleaning task on a flat surface and the generalization on different higher, titled, and curved surfaces that belong to common objects. The lower row shows what the robot perceives of the environment; the blue dots in space are the source distribution, recorded before giving the demonstration (depicted as dashed line), and target distributions recorded before executing the roll-out transported policy (depicted a solid line). Fig. 7.15 also highlights how the roll-outs follow the shape of the surface, showing a successful generalization of the robot position and orientation. As previously stated, no external force-torque sensor is used to adapt the orientation of the end-effector on the tangential direction of the surface. However, an observer of the applied external force between the robot and the surface is estimated from measured torques in the joints, see Appendix A.3. Fig. 7.16, depicts the estimated norm of the force exchanged with the surface, where the same increasing/decreasing trend is captured on the different surfaces.

7.6 LIMITATIONS AND OPEN CHALLENGES

Despite the successful application of the proposed policy transportation on different challenging tasks, we can foresee some limitations and future challenges to improve the applicability and have a broader impact.

For example, we assume knowing the matching between the points of the source and the target distribution. However, in many complex scenarios, this limitation can be problematic, and some different pre-processing algorithms, such as optimal transport [110] or iterative closest point (ICP), need to be adopted to perform the matching. Additionally, semantic matching can increase cross-domain generalization, for example, by adapting the reshelving strategy to a completely different shelf type or adjusting the dressing policy from an adult to a baby arm.

Another assumption of the developed method is dealing with static environments, i.e., the target distribution does not change during the policy's rollout. However, this assumption can fall when dealing with the reshelving of moving objects or when trying to dress real humans that will probably move before and during the interaction. However, supposing to know the state of the target distribution, the nonlinear transportation policy can be updated online by changing the desired deformation labels in the GP. However, the fitting of the transported policy $\hat{f}$ makes it challenging to perform the generalization online.

Finally, given that in complex scenarios, the generalization may be inaccurate, the use of interactive human corrections may increase the resulting manipulation performance [38]. However, changing the generalized policy opens the question of whether interactive corrections should be propagated back to the source policy and how. Additionally, in case many source distributions/policies are recorded, the choice of generating the target policy by transporting all of them, like in Fig. 7.11, or by selecting the best one, according to some similarity criteria, can open exciting developments of the proposed theory.

7.7 CONCLUSIONS

In this chapter, we address the prominent but challenging problem of policy generalization to novel unseen task scenarios. We formulate a novel policy transportation theory that, given a set of matched source and target points in the task space of the robot, regresses the function that, most likely, would match the source and target distribution. Additionally, we showed how the same transportation function and its derivatives can be exploited to transport the original policy dynamics, rotation, and stiffness while retaining uncertainties in the process. The same algorithm, which uses a Gaussian Process at its core, was tested and compared with different state-of-the-art regressors or different generalization methods, showing how, even with only one source demonstration, it results in better or comparable performance. However, the main requirement, for a successful generalization is to track and match important task points in the original scenario, where the demonstration was given, and the corresponding points in the new scenarios.

We validated the proposed approach on a Franka Robot, testing it on three different tasks: product reshelving, arm dressing, and surface cleaning. These various tasks were never tackled together by the same generalization algorithm, and they usually were performed with ad hoc solutions, for example, to keep a constant force when cleaning a surface. Despite this, the proposed policy transportation algorithm performed successfully in all of them. The tracking requirements were satisfied using fiducial markers or directly the point cloud estimated with the infrared camera sensor. Future development will have to focus on scaling the process on big and unmatched point clouds of complex (and deformable) objects to manipulate while allowing the use of human feedback in the fine-tuning of the resulting policy.

8

Learning Interactively to Resolve Ambiguity in Reference Frame Selection

When teaching robots with demonstrations, the choice of the correct reference frame for the learned trajectory could be ambiguous in case there are multiple valid candidates. The algorithm introduced in this chapter, Learning Interactively to Resolve Ambiguity (LIRA), proposes a new interactive framework for solving ambiguity in the demonstration. Teachers do not have to give multiple demonstrations to solve the ambiguity, but their feedback is used to search for the correct solution using a candidate elimination procedure. LIRA is applied for the manipulation of objects using Movement Primitives, where ambiguity typically arises after the segmentation of the trajectory, and the choice of the correct reference frame is not unique. Experiments were conducted with a Franka Emika Panda manipulator for different pick-and-place scenarios. Results showed that the proposed method eliminates the possibility of flawed executions due to ambiguity in the frame selection of a movement primitive. A user study showed a significant reduction in the task load of the user with respect to a system that does not detect ambiguities. A video of the experiments can be found here: `https://www.youtube.com/watch?v=tSQJP8Hpmbk`.

This chapter is based on 🗎 ***Franzese, G.****, Celemin, C., and Kober, J. (2020). Learning Interactively to Resolve Ambiguity in Reference Frame Selection. In CoRL (pp. 1298-1311)* [53].

8.1 Introduction

In Learning from Demonstrations (LfD), the learning agent requires enough representative demonstrations to understand a task's objectives to avoid any possible behavior misassociation. However, the requirement of representative examples could be demanding for a human teacher. Especially for an end-user, it might not be clear how many demonstrations are sufficient. In this work, we focus on creating a system that learns from demonstrations and is ambiguity-aware. Therefore, it can leverage a teacher's corrective feedback to disassociate the multiple interpretations of the demonstrated behavior. With this awareness, the robot can prevent executing wrong (sometimes dangerous) actions, either with active queries before action execution or enabling kinesthetic corrections by the user during the execution of an ambiguous decision.

Ambiguity is an attribute of any idea or statement where the intended meaning cannot be inferred as there are multiple interpretations. Teaching, as a form of transfer of knowledge, and not only of notions (strictly connected to the context), can be ambiguous. For instance, in a teacher-student scenario at school, if the teacher is explaining something that can have different meanings or interpretations, and the exam question is in the same circumstances as the explanation, a simple transcript of the teacher's exact words will lead to the best score. However, the meaning of learning is not repeating the lecture of the teacher by memory; a good exam should check whether the student is able to generalize the concept to different situations. Similar situations may show up when robots learn from demonstrations. For instance, a robot arm is shown to go towards a cup placed on a coaster (see Fig. 8.1), but in a new situation, these two objects are in different positions: Where should the robot go? To the cup or to the coaster? Without any additional information, this ambiguous situation could not be solved [16].

The ambiguity investigated in this chapter prevents the policy from randomly selecting the dependence of the goal on a reference frame whenever the demonstrations are not completely informative. The human teacher observes the current policy and, by interacting with the environment, provides feedback that LIRA employs to update the correct goal. The approach reduces the (unknown) amount of required full demonstrations that a human teacher needs to provide, rather than relying on less demanding interactive corrections, as discussed in [15, 87, 126]. Therefore, the system decreases the learning time, the probability of failure, and the workload of the user.

In the next section, related work on LfD, few-shot learning, ambiguity, and reference frame selection is reviewed. In Sec. 8.3, the LIRA algorithm is explained in detail. Sec. 8.4 shows the experimental validation of this innovative methodology, and finally, Sec. 8.5 concludes the work and describes open challenges.

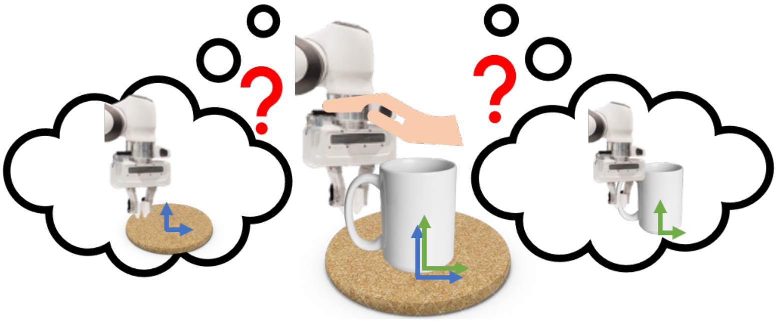


Figure 8.1: ambiguity of demonstration of grasping a cup on a coaster. The two possible hypotheses for generalizing in future segments are illustrated as two conflicting interpretations of the task.

8.2 Ambiguiy in Learning from Demonstration

LfD is an intuitive alternative approach for encoding robot policies instead of programming them by hand. Having the possibility to show the desired behavior to a robot and to give corrections or feedback to the demonstrated trajectory has been shown to be a faster and more versatile methodology, which is user-friendly for adapting policies to multiple scenarios [11]. One of the challenges in LfD is to extract as much information as possible from the demonstrations, avoiding burdening humans with the responsibility of providing several demonstrations. One-shot learning [47] proposes a NN architecture where the new task is learned with only one new demonstration. Alternatively, the works in [147], [92], and [122] use Movement Primitives for learning new tasks in a few shots. All the types of learning approaches underline the possible presence of ambiguity in the demonstrations, but their focus is on different problems.

LfD claims to indirectly transfer the intention of the teacher in solving a task to the robot for a particular application. The mismatch between the human *intention* and robot *deduction* can generate two possible situations: reversible and irreversible ambiguity, see Fig. 8.4. We define a candidate of the learning process as any of the possible deducted ideas that could match the shown human demonstration.

A reversible ambiguity stands for the ability of the robot to select one candidate to resolve the ambiguity. However, this is not always possible. For example, the mixing of demonstration of two different grasping modalities (see Fig. 8.3b) is not solvable with candidate elimination (there is only one deduced candidate), making the situation forever ambiguous. This work focuses on reversible ambiguity. Reversible ambiguous scenarios or demonstrations do not allow a unique definition of the decision directly after the demonstration. However, given a list of possible policy candidates, our assumption is that at least one of them will match the human *intention*.

Ambiguities exist in learning agents due to factors like how the data is used to build the policy, the observability in demonstrations, the quality of recorded data, and the structure of the policy, among others. Below, a classification of the most common types of ambiguities that could exist when a robot is learning from a teacher is introduced.

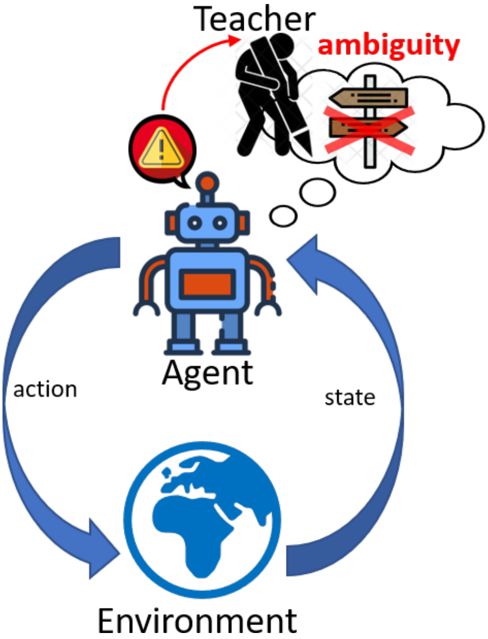


Figure 8.2: LIRA makes the agent ambiguity-aware. When an ambiguous situation is faced, a warning message is sent to the teacher that selects or demonstrates the right candidate and resolves the ambiguity.

8.2.1 Multiple Features Inconsistency

In the situation of Fig. 8.3a, the teacher shows the grasping of a red hat. How should the robot behave in order to match the demonstrated intention? By gripping a red hat? By gripping a hat of any color? By gripping a red object of any type? By gripping any possible object? Thus, a single demonstration maps to multiple different behaviors that the teacher might have intended. This one-to-many (or many-to-many) mapping from demonstration is the concept of the ambiguity approached in [16], where there is the formalization of restricted hypothesis space (for making the ambiguity resolution faster).

8.2.2 Multi Modalities Inconstancy

There are cases where multiple demonstrations contain contradictory actions given the current state. Depending on the regression methods these multimodalities could lead to inconsistent and flawed policies. In Fig. 8.3b, the human is showing how to approach the trophy from each one of the two possible handles. For instance, the use of a simple regression using mean squared error obtains a policy that computes trajectories through the valleys of the demonstration distribution, which is wrong for some applications. The literature is proposing solutions like [141] where a Voronoi tesselation is helping in representing the ambiguity of future steps through multiple hypotheses.

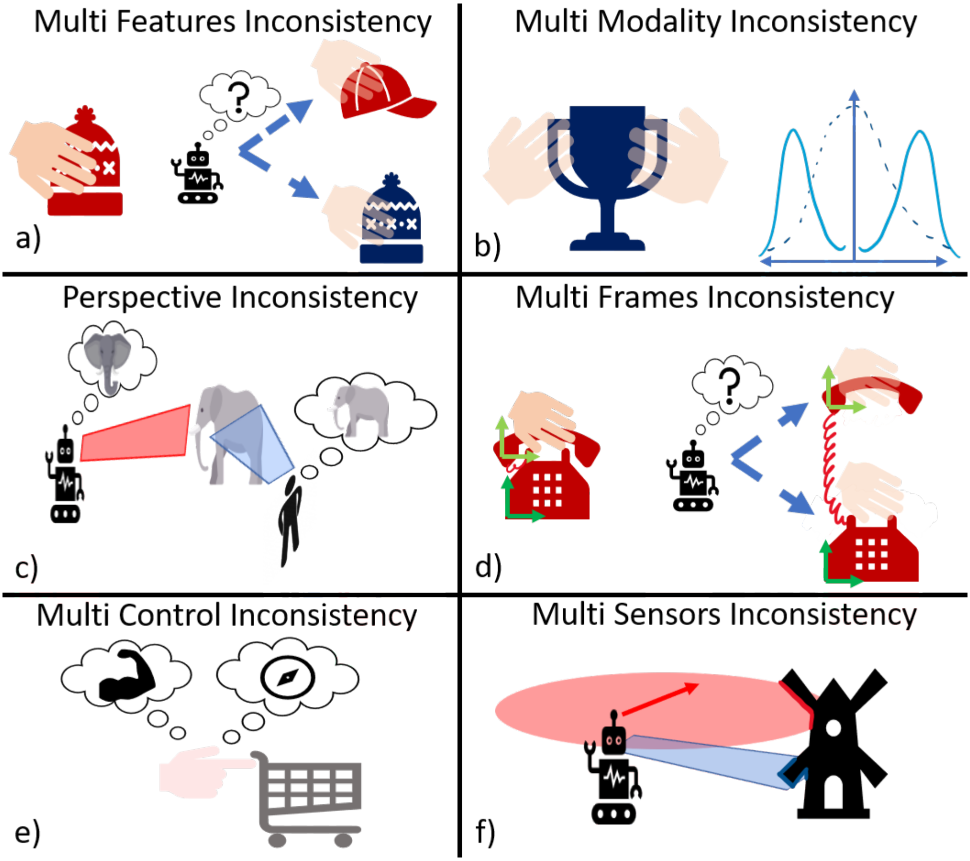

Multi Features Inconsistency
a)
?
Multi Modality Inconsistency
b)
Perspective Inconsistency
c)
Multi Frames Inconsistency
?
d)
Multi Control Inconsistency
e)
Multi Sensors Inconsistency
f)

Figure 8.3: Six different examples of ambiguity.

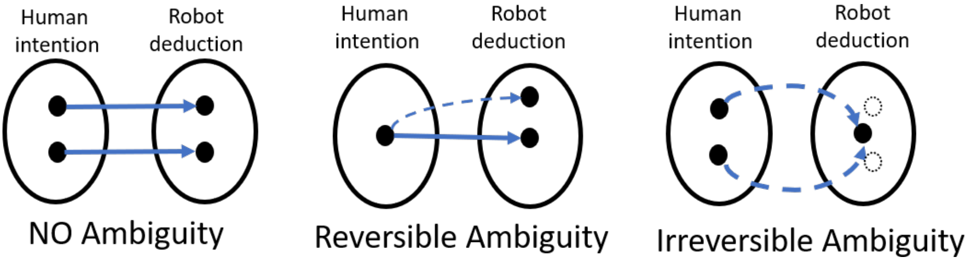


Figure 8.4: Ambiguity in Learning from Demonstrations. In the first case, there is a perfect match between human intention and robot deduction. In the case of reversible ambiguity, the robot is deducing some redundant policy candidates, and the correct one can be found later on with more demonstrations or corrections. Finally, the irreversible ambiguity is due to the impossibility of retrieving the human intention, bringing a wrong result that cannot be fixed with candidate selection.

8.2.3 Perspective Inconsistency

Depending on the sensors' inputs in the perception system of the robot, there could be an observability mismatch between the demonstrator and the learning robot, which is a source of potential ambiguities. The illustration in Fig. 8.3c describes this type of ambiguity also investigated in [24]. In case the demonstration is provided by a naive human teacher, not an expert in machine learning, the result can be sensible from a human's perspective and ambiguous for the learning algorithm point of view, without the possibility to generalize properly. An example of perspective mismatch between the robot and the human is the case in which the human is showing the grasping of an object that does not have any frame associated with it. The result is going to be the inability to generalize the motion: what the user is considering as an object is invisible from a robot's point of view.

8.2.4 Reference Frame Ambiguity

Actions, e.g., trajectories, change according to the observations, i.e., reference frame positions. In trajectory learning, the use of multiple reference frames allows encoding the movement not with respect to the world coordinates but relative to other points/objects, which increases the chance of matching the demonstrator's intention and allows for natural generalization to location changes of the points/objects. The learning algorithms are in charge of building the relations between observations and actions. When actions are associated with more than one of the frames observed by the robot, multiple demonstrations with a rich variety of initial conditions are required; this allows the breaking of false associations with redundant frames. However, this is not obvious for a non-expert user, who does not know the required data to have a representative set of demonstrations. In [28] and [75], a Task Parametrized GMM is used for describing the set of demonstrations respect each of the reference frames. In a new situation, the resulting trajectory is based on the overlap of the GMM components that are moving according to their reference frames. This approach allows to switch the dependence from different reference frames in each movement segment and to regulate the robot stiffness proportionally to the variance of the model. Similarly, in [92] a method for inferring the "correct" reference frame, with the computation of a score for each candidate frame, is proposed. The score is inversely proportional to the combination of the inter-demonstration variance derivative and final goal variance. After the segmentation of the trajectory, the method selects the reference frame with the highest score for each movement segment. Alternatively, in [122] the selection of the frame in each segment is performed by clustering the relative final position (of each one of the demonstrations) with respect to each frame in the world frame, and choosing the frame that owns the biggest cluster. These related works underline the necessity of having more than one demonstration and that those are provided without ambiguities with respect to the way the relevance of each reference frame is computed. This chapter is intended to propose a method that deals with ambiguous situations in reference frame selection and solves that with interactive human feedback.

8.2.5 Multi Control Modality Inconsistency

Demonstrations with manipulators for force interaction tasks record profiles of positions and forces. However, during execution, the robot has to decide which of the profiles to track, this is not always evident from the data on the demonstrations, and could create ambiguous scenarios. In [92], the same principle of minimum variance is used for the selection of the control modality: force or position/velocity control. In the case the choice is equivalent, the decision cannot be arbitrary because in new scenarios the wrong option could be dangerous. In Fig. 8.3e the robot is not sure which modality to follow either the recorded push with the force sensor, or the recorded wheel velocities.

8.2.6 MULTI SENSOR INCONSISTENCY

This category might be overlapping with the Multi Features case, in which the actions could be related to several features, or to information obtained by different sensors that could be complementary, contradictory, or redundant. Similar to the previous scenario, where the multiple sensors are used for making the choice of which control modality to follow, the one illustrated in Fig. 8.3f is a metaphoric illustration for showing the ambiguity that is resulting from multiple sensors. The variance method used in the previous two examples can also be implemented in this scenario. Above, some of the possible categories of ambiguous situations that could be faced when learning from demonstrations with robots have been mentioned, along with some methods that help in dealing with them. Several other research areas also intersect with similar fundamental questions about how to filter relevant information from observations to achieve effective generalization in learning. For instance, State Representation Learning [22, 104], and learning task features from demonstration [40, 115], approach this problem from a different perspective, but not with the focus on ambiguity-awareness for active correction requests, as investigated in this chapter.

8.3 LIRA: LEARNING INTERACTIVELY TO RESOLVE AMBIGUITY

In ambiguous situations, the learning system does not completely understand the intention of the task, therefore it has to randomly/arbitrarily choose one out of all the possible candidate interpretations, e.g., either the coaster or the cup of the example in Fig. 8.1. This makes the robot to behave in an unexpected way, having a negative impact in the user experience, and consequently in the human robot interaction, specifically in factors like engagement, trust or compliance [68]. If users need to be careful about how to demonstrate, the teaching task becomes slower and increases their workload. This creates a user experience in which the robot does not seem to be "intelligent" or to have a certain level of situational "awareness". Without a system that solves ambiguities, a robot learner will request an entirely new demonstration with the right decision, trying to understand how to match the *intention* of the teacher with the robot *deduction*.

With LIRA, illustrated in Fig. 8.2, the robot solves this problem of ambiguous situations by asking the user whether the current choice is correct. If the choice is incorrect, the robot enables the user to guide it, using different possible interaction modalities as shown in the last paragraph of this section. This work assumes that policies are represented with sequences of movement primitives. During demonstrations, the only recorded data is the positions of the EE, and the frames. The output of LIRA is the goal position for each of the movement primitives. LIRA is hence complementary to learning the shape of the movement via many movement primitive representations.

LIRA initially takes the goal position from the demonstrations, and transforms it into the

Algorithm 4 LIRA in Reference Frame Selection

```
Input: Demonstration(s) of the task and a list of valid frame candidates for each movement primitive
for each movement primitive do
    Observe the positions and orientations of reference frames
    Rank the frames based on inter-demonstrations goal variance
    Filter out frames with non-minimum variance using priors
    Project the goal's position with respect to each reference frame in the world frame
    Group goals that overlap in the world frame
    for each group in the list in descending order of dimension do
        Move the robot toward the group's goal
        Ask for human feedback
        if Positive Feedback then
            Eliminate frames outside the group from the list of valid frame candidates
            Break the search and proceed to the next movement primitive
        else
            Eliminate the groups from the list of valid candidates based on the feedback
        end if
    end for
end for
```

local coordinates of each of the n reference frames. The set of all possible single reference frames is

$$F = \{f_i | 1 \leq i \leq n\}$$

and each of its elements f_i has its associated goal ϕ_i contained in the set of goals Φ.

The proposed method is in charge of selecting the right frame f_i for each movement primitive, such that moving in its relative coordinates towards its goal ϕ_i is the correct generalization. Whenever there are ambiguities, the teacher's feedback queried by LIRA helps to reduce the amount of goal candidates, which are linked to the observed frames. The operations that Algorithm 4 is performing in order to take into account multiple demonstrations, priors, overlap of goals and human feedback are explained below.

Candidate ranking: In case there are multiple demonstrations, the generation of the priority list for the reference frame is based on the measure of goal variance. After the segmentation, the final EE position of each movement primitive is computed with respect to each of the n frames, and the standard deviation of the distribution of inter-demonstration goals is computed for each frame in each movement segment. LIRA prioritizes the frames that have lower inter-demonstration variance, as in [92].

In case only one demonstration is provided, the covariance cannot be calculated and a default value is set: only priors and human feedback can be used for finding the right candidate.

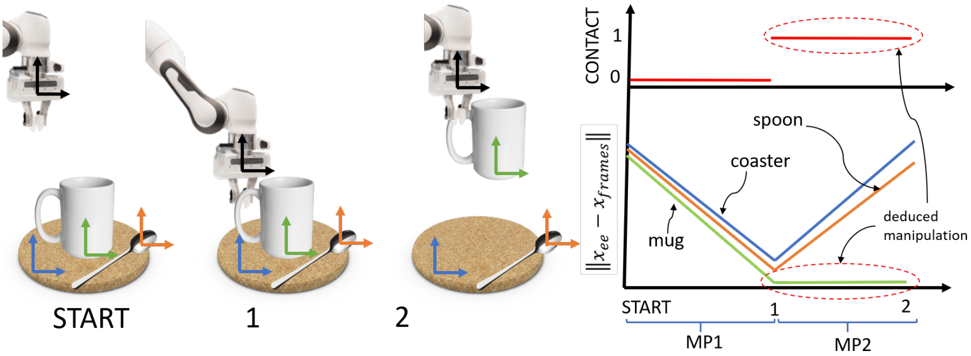


Figure 8.5: Demonstration of the grasping of a cup on a coaster. Because there is only one demonstration, the selection of the correct frame is ambiguous if no priors are used. In MP2, the grippers are closed (recorded contact) and the measure of the distance between the cup and the end-effector is kept constant. LIRA infers that the frame of the cup is being manipulated, giving priority to it in the resolution of the ambiguity of MP1. In the same way, LIRA removes the cup's frame from the list of candidates of MP2 because that goal would be already satisfied at the beginning of MP2.

Reducing ambiguity using priors: As described in [16] and [52], the use of priors reduces the dimension of the candidate frames search space, making the ambiguity resolution faster. Various priors can be included to prioritize frames in the selection, e.g., distance to the frame origin (as a proxy for an object/goal being reached) [7] or temporal consistency (discouraging or encouraging frequent switches between frames). In some cases, the selection of the goal of each movement primitive is based on information of the immediate subsequent segments.

For example, suppose a constant grasping contact and a constant distance with respect to a frame is detected in an entire segment. In that case, it means that the frame (i.e., the object it is attached to) is being manipulated. Hence in the preceding segment, in case of ambiguity, LIRA will give priority to that frame, encoding that an object needs to be grasped before being manipulated. An example of this situation is illustrated in Fig. 8.5.

Similarly, the frame being manipulated can be discarded as candidate reference frame for that segment. This frame is temporarily rigidly attached to the end-effector, and the goal for that frame would hence already be satisfied at the beginning of that segment.

Candidate grouping: After applying the filtering with the priors and selecting the frames with the lowest inter-demonstration variance (in the limit of a threshold), LIRA groups the reference frames according to their current goal positions in the executing moving primitive. The j^{th} group $G_j \subset F$ is defined as a subset of reference frames whose associated goals are overlapping in the global coordinates. Therefore, regardless of the reference frame chosen among the group, the movement primitive would have the same global goal position. Thus, for a specific situation, there is an ambiguity with an amount of c candidate goals, where c is the amount of groups encountered by LIRA. Goals are grouped by the distance between the candidate goals, and a threshold defining a tolerance region. In the example of Fig. 8.6, the goals of the three frames are organized in two groups, given the drawn tolerance region. Therefore, in this ambiguous situation, LIRA gives priority to the biggest group on the left (line 8 of Algorithm 4).

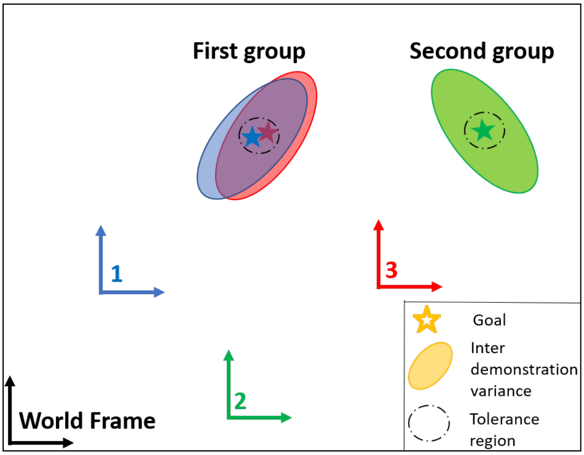


Figure 8.6: Candidate Grouping: all the goal distribution have the same inter-demonstration variance and they are not filtered away by any prior. Then, LIRA creates groups of goal, checking if they are overlapping within a tolerance.

When the user is labeling one of the groups as correct, all candidates in the other groups are eliminated. All members of the selected group are saved to the memory and become valid candidates for future iterations. Due to the human feedback, there is no longer an ambiguity about the goal in this specific situation. Nonetheless, those multiple selected candidates might result in an ambiguity in future situations if the goals of the group do not overlap anymore.

Human feedback: LIRA moves the robot towards the goal of largest group, i.e., maximizing the probability to find the right frame. Then it queries feedback from the teacher (line 10 in Algorithm 4) in these three possible ways:

- *Labelling correctness (right/wrong)*: The algorithm asks whether the current goal that was reached is correct or not. This query could be done through a graphic interface, or with a sound signal, whereas the teacher's response could be obtained with a keyboard, remote control, voice, etc. When the feedback is negative, in the **Eliminate** step (line 15 of Algorithm 4), LIRA only discards the frames belonging to the current selected group.

- *End-effector directional kinesthetic perturbation*: In order to avoid the dependency on a human-computer interface, another proposed solution is to make the robot compliant and to enable the human to correct the candidate choice simply by pushing the robot towards the direction of the correct goal (Fig. 8.7.2.b,1), i.e., LIRA assumes positive feedback (line 11 in Algorithm 4) when there is no kinesthetic perturbation, and negative otherwise. This interaction mode is richer in information than the previous one, as in case of a wrong guess of reference frame, the user interaction does not only make LIRA discard the wrongly selected goal, but also all the potential candidates which are in the opposite direction of the correction, therefore, the ambiguity resolution would be faster.

- *End-effector kinesthetic movement*: Similarly, in case of ambiguity, LIRA allows the user to move the EE close to the right goal (Fig. 8.8.2.b,1). If no correction happens, LIRA assumes positive feedback (line 11 in Algorithm 4). Otherwise, in case of EE movements, it assumes negative feedback (line 14-15) and labels the closest group's goal to the final EE position after the interaction as correct, discarding all the others. This feedback modality is the most information-rich and should be considered the best option when there are many perceived reference frames. However, due to the longer interaction with the user, this option would be expected to be more mentally and physically demanding compared to the other two options.

8.4 Experiments and Results

LIRA has been proposed to solve the ambiguity in the selection of reference frames in an interactive way, reducing the total number of required full demonstrations. Experimental evaluations have been carried out to measure the effectiveness of LIRA. Three validation scenarios of pick and place tasks and a user study were run in order to illustrate the operation of the system and to evaluate its performance compared to a system that is not ambiguity-aware.

All the experiments were conducted with a 7 DoF Franka Emika Panda robot arm with a parallel gripper. The robot is controlled with Cartesian impedance control, which allows to have compliance for the interactions with the user. The robot is used in gravity compensation mode for recording the kinesthetic demonstrations. The demonstrated trajectories are segmented in movement primitives, and represented with linear attractors to their goals, with constant velocity. For the validation of LIRA, only one fully kinesthetic demonstration is provided as input to the policy.

8.4.1 Validation Tasks

Fruit sorting in crates: In the case depicted in Fig. 8.7, there are apples and cucumbers in the workspace that need to be picked up and placed in their respective crates (on the left).

In the sequence shown in the top row (1.a-b) of Fig. 8.7, the scenario of the demonstration is reproduced, whereas in the bottom row (2.a-b,1-2), the position of the cucumber is different and the position of the crates is swapped (the scenario differs from the demonstration and it is potentially ambiguous as a consequence). Due to the priors, the robot is able to go and pick up the cucumber in the new scenario, however, it requires the disambiguation for the goal of the second segment of the task (selection of the basket for placing the cucumber). Therefore, the robot moves towards the basket of apples but stops to request the teacher's feedback before placing the object. Then, the simple physical perturbation towards the right goal is enough to solve the ambiguity. Without the ambiguity-awareness system, the user would have to demonstrate the whole trajectory in the second scenario (after observing a flawed object placement), instead of giving one correction during the autonomous execution.

Another validation example is depicted in Fig. 8.8, where the user is showing the restocking operation of bananas in the crates of a supermarket. After one demonstration (1.a-b), when the environment is modified (2.a), LIRA faces an ambiguity in the selection

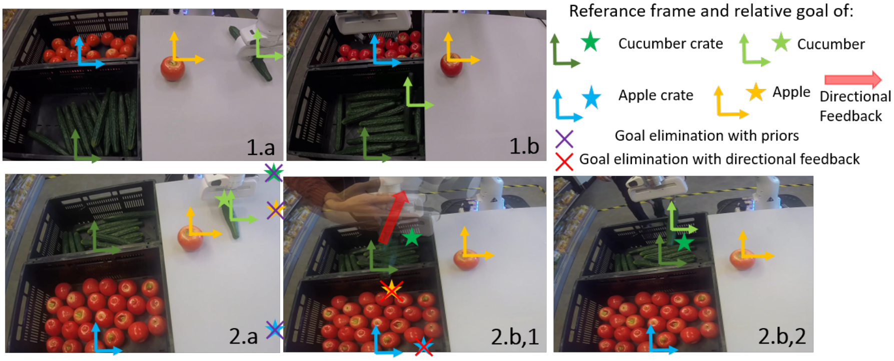

Referance frame and relative goal of:
Cucumber crate
Cucumber
Apple crate
Apple
Directional Feedback
Goal elimination with priors
Goal elimination with directional feedback
1.a
1.b
2.a
2.b,1
2.b,2

Figure 8.7: Fruit pick and place 1. The first row shows the policy execution in the same scenario of the demonstration (1.a-b), i.e. no ambiguity can arise. After a frame rearrangement, in the second row, LIRA uses the manipulation prior for the picking operation (2.a) but it asks the human feedback (2.b,1) to solve the ambiguity in the placing one (2.b,2).

of the goal for the banana placement (2.b,1). As LIRA is implemented with the kinesthetic movement feedback in this example, the robot becomes compliant on the EE and the user moves it towards the right goal (2.b,1), solving the ambiguity (2.b,2).

Box stacking The third validation case is a box stacking task, as shown in Fig. 8.9. The sequence of pictures on top shows the demonstrated scenario, where the stack of boxes is organized with the green one on the bottom, the orange in the middle, and the black one on top, each of which is perceived as a reference. For the second scenario (the bottom sequence of pictures), the use of the manipulation prior eliminates ambiguities during the grasping segment. However, in the placing operation, an ambiguity is found when the user disturbs the environment. With this disturbance, the goals associated with the frames on the orange and green boxes do not overlap anymore. With the information obtained with the first demonstration, the robot learner faces an ambiguous situation, in which the goal for placing the black box could be with respect to either the orange or green box. The physical correction when the user pushes the end-effector makes the robot to understand that the black box's position should be with respect to the green one.

8.4.2 User Study

The performance of LIRA has been evaluated and compared with a system that learns from kinesthetic demonstrations without ambiguity-awareness: in the case of frames with the same inter-demonstration variance, the selection is random and, if wrong, it requires a new complete kinesthetic demonstration. The intention of this study is to compare the amount of time reduction by the proposed interactive system for this specific task, along with the workload required from the user, both, with respect to the system without ambiguity-awareness. In this regard, a user study was conducted with twelve participants, who had to interact with both of the aforementioned systems, for teaching the same task. The type of feedback used in LIRA for the experiments was the directional perturbation because it was considered to be the best compromise between efficiency and user-friendliness.

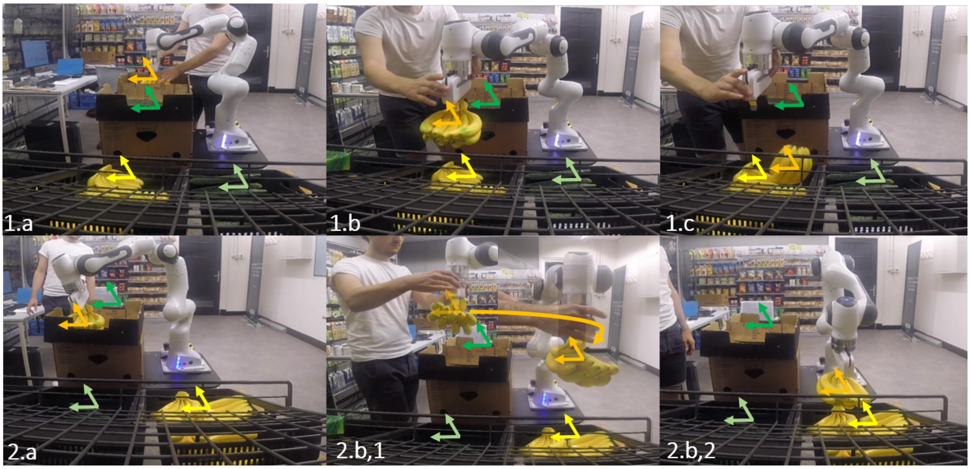


Figure 8.8: Fruit pick and place 2. The first row shows the user demonstrating the task with Kinesthetic Teaching (1.a-b-c). After the frame rearrangement, LIRA uses the manipulation prior for the picking operation (2.a) but it asks for feedback in the placing one (2.b,1) allowing the user to move the EE close to the desired goal (2.b,2).

The participants were students of an engineering faculty who had no prior experience of kinesthetic teaching, and their ages ranged between 22 and 30 years. In order to have a measurement of the user workload during the teaching process, the NASA Task Load Index (NASA-TLX) questionnaire [69] was performed after the participants finished interacting with the system. This questionnaire has six questions related to mental demand, physical demand, temporal demand, performance, effort, and frustration. The task was about stacking pairs of boxes with a predefined order. There were six boxes numbered from 1 to 6 on a table, and the objective was to place box number 1 on top of 2, 3 on top of 4, and 5 on top of 6. At the end of the execution of each scenario, the objects were rearranged according to a predefined order.

In the experiment with only kinesthetic teaching, the robot recorded the first demonstration, and then it was tested for the other cases. However, whenever the system failed, it would receive a full kinesthetic demonstration of the correct execution of the task for that scenario from the user. After 4 demonstrations, all the users managed to teach the task successfully, i.e., the ambiguities were solved. For evaluating LIRA, a full kinesthetic demonstration was required only for the first scenario. For the other scenarios, LIRA requested the user to push the end-effector towards the right direction of the goal, in case of a detected ambiguity.

The results of the experiments showed the learning time was reduced by at least a half with LIRA, since it allows corrections during execution time, and does not request the user to record entire new trajectory demonstrations. In each scenario, LIRA never performed a wrong task reproduction, as the ambiguity-awareness prevented the robot from executing a mistaken decision with the support of the human teacher.

The video of the experiments shows the teaching of the task of placing boxes 1 on 2, 3 on 4, and 5 on 6 when the only recorded data is the end-effector position with respect to each of the reference frames (attached to each box). As depicted in Fig. 8.10, in case of ambiguity, LIRA creates groups and uses the positive/negative feedback of the human for selecting the

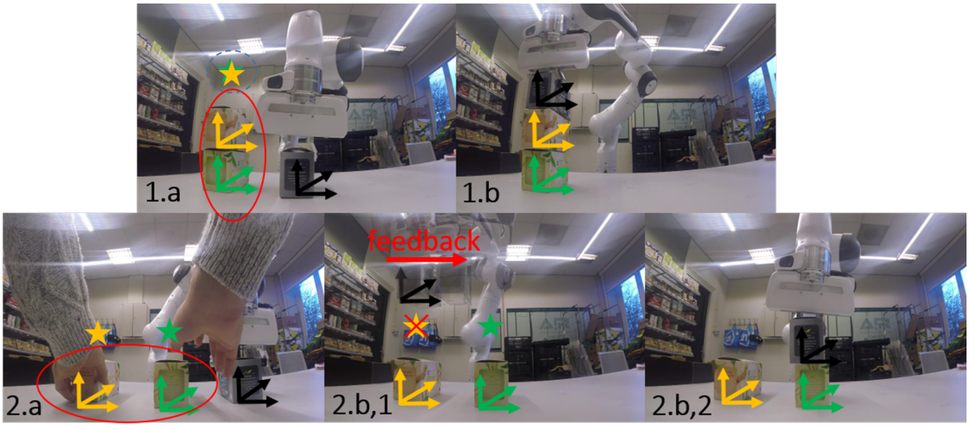


Figure 8.9: Frames overlap ambiguity. The first row shows the demonstration (1.a-b). Because the green and yellow boxes are stacked, then the relative position of their frames is never changing in new scenarios and their goal will always be grouped in the placing segment of the black box (1.b). When the overlap is broken (2.a), an ambiguity is faced and LIRA asks for feedback (2.b,1): LIRA infers the user's preference of the dependence of the goal on the green frame (2.b,2).

right candidate. The manipulation prior (introduced in this chapter) automatically solves the ambiguity in the picking operations (MP 1,3,5) of the boxes 1,3,5. Similarly, in the placing operations (MP 2,4,6) the prior deletes the current manipulated frame, respectively 1,3,5. However, with only one full demonstration in Scenario 1 (Fig. 8.11) and with the use of priors, there is still not enough information for uniquely finding the dependence of the goal

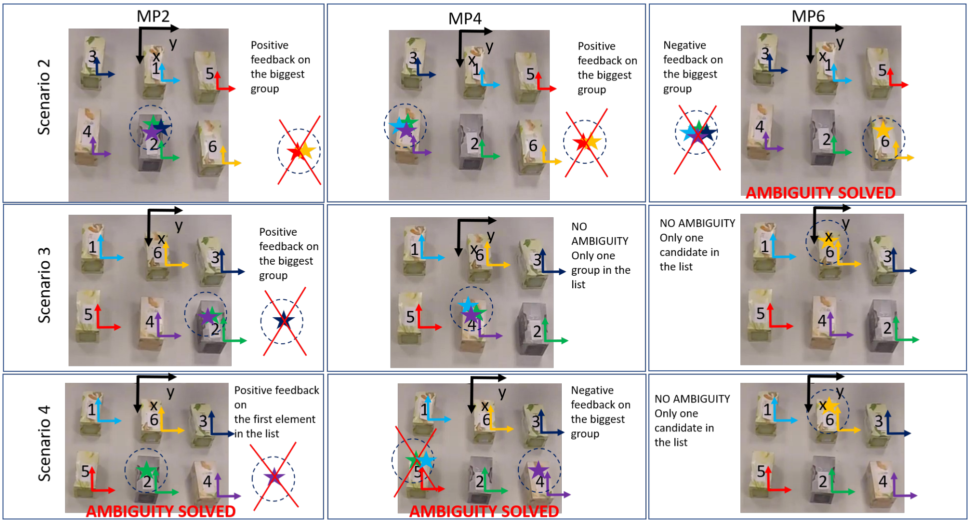

MP2
MP4
MP6
Scenario 2
Scenario 3
Scenario 4
Positive feedback on the biggest group
Positive feedback on the biggest group
Negative feedback on the biggest group
AMBIGUITY SOLVED
Positive feedback on the biggest group
NO AMBIGUITY Only one group in the list
NO AMBIGUITY Only one candidate in the list
Positive feedback on the first element in the list
Negative feedback on the biggest group
NO AMBIGUITY Only one candidate in the list
AMBIGUITY SOLVED
AMBIGUITY SOLVED
x
y

Figure 8.10: Details on the disambiguation procedure with LIRA and the human feedback in the Scenarios 2, 3, 4 after one complete Kinesthetic Demonstration in the Scenario 1 of Fig. 8.11

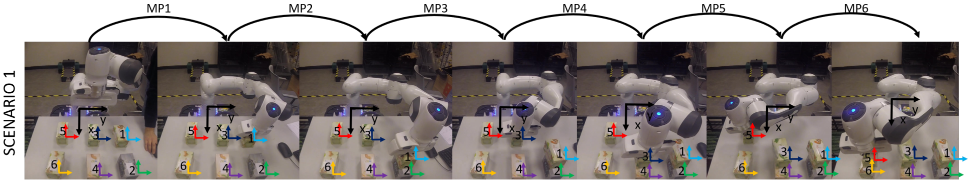


Figure 8.11: User study frames of the Kinesthetic Demonstration in the first scenario. The teaching task is stacking box 1 on 2, 3 on 4, and 5 on 6. The task is segmented in six movement primitives: three picking operations and three placing operations.

from a single reference frame in new scenarios for MP 2,4,6.

In Fig. 8.10, the position of the goals with respect to each reference frame, learned in the first demonstration, are projected in the new scenarios for each of the 'placing' movement primitives, i.e., where there are multiple valid candidates in the list and a potential ambiguity. The groups are formed according to the grouping operation explained in the Sec. 8.3. LIRA is always going to the biggest group first and if the user does not give any perturbation to the end-effector within a time limit[1], it is labeled as correct, i.e., all the elements of other groups are removed from the list of valid candidates. Alternatively, in case of a perturbation, the selected group is labeled as wrong and all its elements are removed from the list of valid candidates. The different rows of Fig. 8.10 show how the user, through positive and negative feedback, removes all the redundant candidates from the list until there is only one candidate left for each MP and no ambiguity can arise in future different scenarios.

Fig. 8.12 reports the results of the NASA-TLX questionnaire, which show how demanding the teaching task was with each of the methods. It is possible to observe that for all the questions, the participants reported better results with LIRA. For all the questions, the difference of the mean scores are considerable, and with LIRA the variance is lower in general. This was expected since the interactive system with ambiguity-awareness eliminated the flawed executions, along with the need of entire new kinesthetic demonstrations, which took about two minutes each time. Rather, LIRA requested shorter interactions of corrective feedback that required only approximately one second of physical contact with the robot.

[1] Please notice the pause in the execution of the task (in the video) when LIRA is unsure about the correctness of the goal and waits for the eventual feedback of the user with EE perturbation. LIRA labels that goal as correct if no feedback occurs.

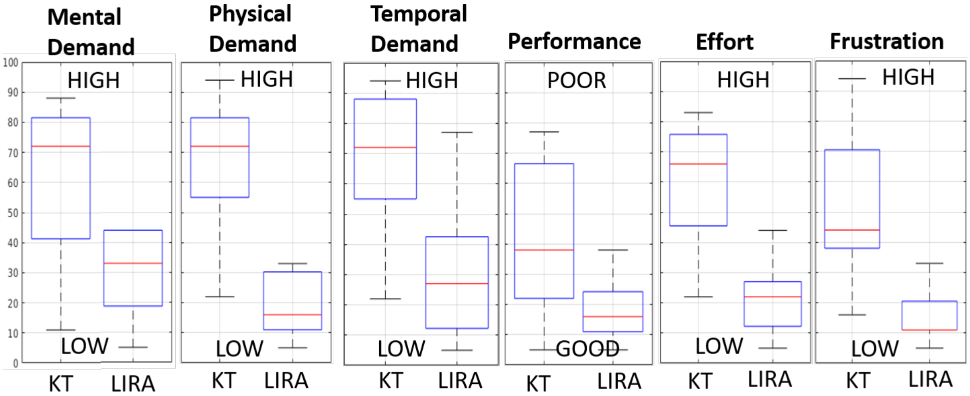


Figure 8.12: Statistical results of the workload NASA-TLX questionnaire that compares Kinesthetic teaching (KT) with LIRA on different aspects.

8.5 Conclusion and Future Work

A system that is able to give awareness to the robot about ambiguities in the selection of reference frames for the goal of a movement primitive has been developed. This awareness is useful for implementing active robot learners, which can prevent the execution of mistaken actions produced by multiple potential interpretations of the demonstrated examples. The robots are able to request local physical corrections or signals of (dis)approval from the user in order to feed its knowledge and eliminate the wrong associations. With this interactive approach, the workload of the teachers is reduced since the user-robot interaction is decreased at both cognitive and physical levels. This approach was validated with a user study, wherein the obtained results have shown an improvement in the user experience, with respect to a system that does not have ambiguity-awareness and an interactive disambiguation modality. This contribution intends to have more friendly robots that are able to share spaces and activities with their end-user counterparts. These robots need to be more and more adaptable to people who are not robot experts or do not have a technical background since they will become part of our daily lives, supporting basic activities.

9 Conclusion

This thesis has investigated how non-expert users can teach a robot to manipulate objects through the proposed interactive imitation learning algorithms. The main characteristics of uncertainty quantification and exploitation made it possible to conclude that uncertainty is necessary for robots to learn from humans and enhance the resulting autonomous performance. This chapter will give more details on the conclusion and insights that can be taken from the reading of this thesis and answer the research questions introduced in Chapter 1. It will also foresee the next challenges and opportunities for learning manipulation skills from human teachers.

9.1 REFLECTIONS

The core objective of this thesis was to devise new algorithms with the singular aim of accelerating the robot's acquisition of manipulation skills, requiring fewer interactions with human teachers, and instilling the capability for uncertainty-aware behaviors in both the learning and execution phases.

The challenge lies in teaching robots contact-reach manipulation skills, such as table cleaning, (un)plugging a socket, or picking up an object at non-zero velocity—tasks that come naturally to humans but prove intricate for robots. The knowledge transfer from human to robot demands numerous interactions to rectify mistakes, enhance execution speed, or improve accuracy. Traditional LfD, in its classical interpretation, falls short as it does not facilitate the incorporation of new data during the robot's execution. Consequently, Ch. 2 introduced readers to the different feedback modalities of Interactive Imitation Learning, given the thesis's contribution to this topic.

However, in interactive learning, quantifying and exploiting uncertainties is still unclear; in particular, quantifying epistemic uncertainty due to the lack of data provided as a demonstration is one of the biggest challenges. Nevertheless, Gaussian Processes, introduced in Ch. 3, is a successful mathematical framework for learning non-linear policies while estimating the uncertainty of the prediction. The chapter tackles and formalizes how statistical models can enhance data efficiency with novel update rules and defines an uncertainty-aware aggregation rule by aggregating new data only if necessary. To address the first research question on the practical implications and performance improvements arising from integrating uncertainty quantification/exploitation and definition of priors, we can highlight that

1. the proposed update rules proved instrumental in modifying the desired attractor and stiffness behaviors, as discussed in Ch. 4, and it enables interactive adjustments to motion speed while correcting gripping and attractor behaviors for executing picking tasks at non-zero velocities, as detailed in Ch. 5.

2. the minimization of the uncertainty generates an attractive force field that effectively guides the robot toward regions closer to the demonstrations, enhancing stability in learned dynamics. Various use cases were examined with the proposed approaches, demonstrating the robustness of learned dynamics when subjecting the robot to disturbances, such as physically halting its motion, as detailed in Ch. 4, 5 and 6.

3. played a key role in regulating the robot's stiffness as a function of the uncertainty, enhancing compliance when human disturbances were introduced to the system, as detailed in Ch. 5 and Ch. 6.

4. the extrapolation-free prior also allowed to safely control the robot orientations, as detailed in Ch. 5 and to attract to the goal of the closest point in the demonstration, promoting local motion stability, as detailed in Ch. 6.

5. the GPs are the only model to retain calibrated uncertainties and provide the uncertainty on the derivative of the deformation map when learning the motion generalization as detailed in Ch. 7.

To study the interactive feature of the proposed machine learning algorithms and answer the second research question on the democratization of robot teaching, in Chapters 4, 5, 6, and 8, non-expert users were engaged in teaching manipulation tasks, including activities like plugging or picking at non-zero velocity, sorting objects or perform a bimanual picking. Despite the users' lack of familiarity with the teaching framework and interface, the users consistently succeeded in instructing the robot.

The performed user studies concluded that

1. when relying solely on kinesthetic demonstration, individuals often struggle to achieve accurate prediction in insertion tasks, even when the demonstration is performed slowly (detailed in Ch. 4), or to achieve the desired (fast) execution time, even when the demonstration is performed quickly (detailed in Ch. 5). However, when corrections to the motion dynamics are provided, individuals can surpass their demonstrated capabilities and achieve the necessary precision or execution speeds that were previously unattainable.

2. when performing kinesthetic corrections, including adjustments such as synchronizing the movements of the two arms or altering the bimanual motion to pick an object from a different location, users typically favored corrections over re-demonstration due to its efficiency and the decreased cognitive burden of controlling two arms simultaneously, as detailed in Ch. 6.

3. non-expert users preferred giving feedback by simply pushing the robot in the correct direction when teaching a long- sequence of tasks rather than aggregating more kinesthetic demonstrations. This preference stemmed from users feeling less physically and mentally stressed during the correction phase compared to repeatedly teaching the task to eliminate ambiguity in selecting the necessary frames for each segment, as detailed in Ch. 8.

9.2 DISCUSSION AND OUTLOOK

While all the algorithms were successfully tested on a real robot with non-expert users, many experiments made strong assumptions about the known localization of objects using tracking systems or markers. The primary goal was to validate and test algorithms without concern for localization uncertainties, allowing for a controlled assessment of the proposed methods.

However, it's important to acknowledge that no vision system can achieve zero uncertainty, presenting a noteworthy challenge in scenarios involving insertion tasks and picking at non-zero velocity. While the current implementation exhibits robustness to small inaccuracies in object localization, the thesis did not explicitly study localization uncertainty, potentially limiting the deployment of the algorithm to unstructured environments. Collecting more demonstrations might not necessarily enhance the robustness of any algorithm to localization errors since aleatoric uncertainties do not depend on the volume of data collected. Nevertheless, particularly for picking, the manipulation strategy can mitigate the effect of the inaccuracies in localization.

This thesis explored interactive learning strategies using various feedback interfaces, including teleoperated devices such as 3D mouse corrections, joypad corrections, and kinesthetic corrections. However, determining which interface is more effective from both human and robot perspectives is not straightforward. Through multiple user studies, the thesis identified that kinesthetic correction can be more intuitive as individuals physically move the robot when providing corrections. However, incorporating kinesthetic correction into the learning algorithm introduces complexities, as distinguishing between a disturbance and required corrections becomes ambiguous. On the other hand, using a 3D mouse to provide corrections in both Cartesian direction and orientation was found to simplify the detection and quantification of feedback from an algorithmic perspective.

Ultimately, the choice of feedback interface involves a trade-off between intuitiveness and algorithmic simplicity, and the effectiveness may depend on specific application requirements and the user's comfort and expertise with the chosen interface. For example, when teaching the robot to interactively increase its velocity at different points in space, clicking a button to speed up the motion locally is often more practical than attempting to physically push the robot. Plus, as tasks involve higher speeds, physical interaction with the robot becomes less safe. It becomes evident that there is no one-size-fits-all ideal user interface. Future developments in interactive learning algorithms should be inherently multimodal, allowing users to choose their preferred teaching modality, with the algorithm adapting accordingly. Each user's feedback preference may vary, with some providing evaluative feedback and others corrective feedback, thereby complicating the update rules of interactive learning algorithms. Despite this complexity, the probabilistic nature of the Gaussian Process framework developed in this thesis holds promise and could serve as a valuable choice for accommodating diverse user feedback modalities.

However, when attempting to teach more intricate manipulation tasks where the user needs to instruct and correct specific hand poses and gripper configuration, limitations in learning from demonstration can emerge. For example, controlling a complex robotic gripper with numerous degrees of freedom using a teleoperation interface is impractical, and there may be no straightforward correspondence between human hands and robot hands, making hand tracking unfeasible. This example may justify the development of a complex hand-like gripper, which is challenging to control using optimal planners or reinforcement learning algorithms due to the curse of dimensionality; still, it could offer a teaching advantage from a broad pool of available human demonstrators. Moreover, the availability of Cartesian control can improve knowledge transfer among different robots. The emphasis, therefore, should not solely be on designing the best-performing gripper or arm design in isolation but rather on developing designs that enhance the transfer of knowledge from human teachers to robots and between robots.

Furthermore, additional research should be directed towards bimanual manipulation, as many existing state-of-the-art teaching methods may not easily scale when instructing two arms simultaneously. From a human-centric standpoint, having dual arms similar to humans can significantly broaden the range of tasks a robot can learn and perform. This eliminates the need for humans to reevaluate how they transfer skills, typically executed with two arms, to a robot equipped with only one arm, such as handling boxes or pieces of luggage.

In addition, humans naturally utilize the complete embodiment of their arms in daily tasks, such as opening doors while hands are occupied. Teaching lower-level policies for the robot's elbow while maintaining specific constraints on the other arm poses intriguing challenges for the manipulation community. Addressing these challenges contributes to advancing robot capabilities and generates enthusiasm for deploying more sophisticated robots to assist in daily activities and handle mundane tasks.

In conclusion, this thesis underscores the significance of uncertainty estimation and minimization for steering robot behavior toward the demonstration distribution and for gauging uncertainty when generalizing policies to new task configurations. Despite the strides made in machine learning, particularly with the success of deep neural networks in handling high-dimensional inputs for complex control tasks, the reliability of uncertainty estimation solutions remains unsolved. Conversely, statistically grounded approaches like Gaussian Processes have proven valuable tools for interactive learning, offering well-calibrated and unbiased uncertainty estimates. Moving forward, the robotics community must explore the effectiveness of both deep learning and Gaussian Process methods in scaling robot tasks to high-dimensional inputs, such as images or graphs, while ensuring accurate uncertainty estimation. Leveraging uncertainty as a safety mechanism to halt the robot or adjust its velocity and stiffness in uncertain situations could prove to be a pivotal feature for the secure deployment of technology, marking a crucial juncture for the future of robotics.

A

Cartesian Impedance Control of Redundant Manipulator

Impedance control is crucial in robot manipulation because it enables robots to interact with their environments in a flexible, adaptive, and safe manner. Traditional position or force control methods can be too rigid, resulting in failures or damage when dealing with unpredictable or dynamic environments. In contrast, impedance control allows robots to modulate the relationship between applied force and the resulting motion, mimicking human-like adaptability in tasks requiring both precision and delicacy, such as assembly, object handling, or interaction with humans.

By dynamically adjusting to changes in the environment—like variable stiffness, damping, or external forces—robots can maintain stability, enhance performance in uncertain situations, and handle delicate objects without causing damage. This adaptability makes impedance control particularly valuable for tasks that involve physical contact, ensuring safer human-robot collaboration and more effective manipulation of complex objects.

Impedance control is one of the main components of this thesis, and this appendix summarizes the fundamental control equations that were used to control the robot in a single or dual fashion. The implementations for the Franka Emika Robot impedance control can be found at https://github.com/franzesegiovanni/franka_human_friendly_controllers and for the bimanual impedance controller at https://github.com/franzesegiovanni/franka_bimanual_controllers.

A.1 Cartesian Impedance Control

The dynamic equation of a robot manipulator is defined according to

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathcal{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \mathcal{G}(\boldsymbol{q}) = \boldsymbol{\tau}_{task} + \boldsymbol{\tau}_{NS} + \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_{friction}$$

where, in order from left to right, there are the mass, the Coriolis and the gravitational term that depend on the joint configuration $\boldsymbol{q}$ and, on the right, the torque for the Cartesian (or task) control, the nullspace control torque, and finally the externally applied torques and the torques generated by the friction in the joints. The task space torque is defined as

$$\boldsymbol{\tau}_{task} = \boldsymbol{J}^{\top}(\boldsymbol{\mathcal{K}}(\boldsymbol{x}_{goal} - \boldsymbol{x}) - \boldsymbol{\mathcal{D}}\dot{\boldsymbol{x}}) + \mathcal{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \mathcal{G}(\boldsymbol{q})$$

where $\boldsymbol{J}$ is the geometric Jacobian, and the stiffness $\mathcal{K}$ and the damping $\mathcal{D}$ give the compliant behavior with a critically damped response [6].

Damping Design

The damping matrix can be designed to simulate a critical damping system [41]. In the framework adopted in this thesis, after computing the orthogonal decomposition of $\mathcal{K}$, i.e., $\mathcal{K} = \boldsymbol{R}\tilde{\mathcal{K}}\boldsymbol{R}^T$ [1], where $\tilde{\mathcal{K}}$ is a diagonal matrix, then $\tilde{\mathcal{D}} = 2\tilde{\mathcal{K}}^{1/2}$ and $\mathcal{D} = \boldsymbol{R}\tilde{\mathcal{D}}\boldsymbol{R}^T$. Please notice that $\tilde{\mathcal{K}}$ is a diagonal matrix, hence the square root is applied to every element of the diagonal.

Nullspace Control

When working with a redundant manipulator, null-space control can be formulated as a projection of the joint impedance in the kernel of the end effector's Jacobian [6], such that the torque in the joint impedance control will have no Cartesian component forces, i.e.

$$\boldsymbol{F}_{NS} = \boldsymbol{J}^{\top +}\boldsymbol{\tau}_{NS} = 0, \tag{A.1}$$

hence given any joint torque, for example generated from impedance in joint space, the nullspace requirements is satisfied if

$$\boldsymbol{\tau}_{NS} = (\boldsymbol{I} - \boldsymbol{J}^{\top}\boldsymbol{J}^{\top +})\boldsymbol{\tau},$$

this definition of $\boldsymbol{\tau}_{NS}$ will make the null space constraint of Eq. (A.1) to be satisfied, no matter the value of $\boldsymbol{\tau}$.

We can define the null space joint impedance control as

$$\boldsymbol{\tau}_{NS} = (\boldsymbol{I} - \boldsymbol{J}^{\top}\boldsymbol{J}^{\top +})(\boldsymbol{\mathcal{K}}_{NS}(\boldsymbol{q}_{goal} - \boldsymbol{q}) - \boldsymbol{\mathcal{D}}_{NS}\dot{\boldsymbol{q}}) \tag{A.2}$$

where $\boldsymbol{\mathcal{K}}_{NS}$ is the null space joint stiffness matrix, $\boldsymbol{\mathcal{D}}_{NS}$ is the null space joint damping matrix, and $\boldsymbol{q}_{goal}$ is the desired configuration. This creates a lower priority on the desired joint configuration that will only generate joint transitions that have a minimal effect on the imposed end effector dynamics.

[1] $\boldsymbol{R}$ is an orthogonal matrix, hence $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$

Joint limit repulsion

Additionally, we can project in the nullspace a joint limit repulsion torque to minimize the risk of hitting a joint limit while minimally affecting the end-effector impedance behavior. We define $\boldsymbol{q}_{safe}$ as the joint configuration with a safety distance from the joint limit $\boldsymbol{q}_{lim}$. In case the robot joint configuration $\boldsymbol{q}$ surpasses the safety limit $\boldsymbol{q}_{safe}$, a null space controller is activated with $\boldsymbol{q}_{goal} = \boldsymbol{q}_{safe}$ only for the joints which are beyond their safety limit. Moreover, the stiffness and the damping for the joint limit rejection are non-zero only when the robot goes beyond the safety limit. It is important that this limit is set before the real-joint limit, otherwise the robot's safety control will lock the joints.

The projection of joint limit torque in the null-space of the Cartesian control generates transitions in the robot configurations that move every joint away from its limit while not interfering with the desired pose control that is actuated in the Cartesian space. This simple solution allows the robot to physically converge to kinematically feasible solutions (when existing) without the need of any inverse kinematics controllers or planning which can be unreliable and converge to local minima.

Safety Attractor and Stiffness Saturations

In order to enhance safety when interacting with humans [67], it is necessary to saturate the attractor displacement and the stiffness to a maximum safe value. To help the definition of the bounds, we can compute them as a function of the desired maximum free-movement velocity ($\boldsymbol{v}_{max}$) and maximum applicable static force of the end-effector ($\boldsymbol{F}_{max}$) (in absolute values).

First, we compute an upper bound for the maximum displacement. Let us consider the dynamics equation of the manipulator in the task space and with the Cartesian impedance control active, i.e.,

$$\Lambda(\boldsymbol{q})\ddot{\boldsymbol{x}} = \boldsymbol{\mathcal{K}}\Delta\boldsymbol{x} - \boldsymbol{\mathcal{D}}\dot{\boldsymbol{x}} + \boldsymbol{f}_{ext};$$

when the robot is in free-movement, i.e., $\boldsymbol{f}_{ext} = 0$, the maximum velocity happens for $\ddot{\boldsymbol{x}} = 0$, that is to say:

$$\boldsymbol{\mathcal{D}}|\dot{\boldsymbol{x}}| = \boldsymbol{\mathcal{K}}|\Delta\boldsymbol{x}|.$$

Thus, given the current setted stiffness $\boldsymbol{\mathcal{K}}$ and the desired max allowed velocity $\boldsymbol{v}_{max}$, $\Delta\boldsymbol{x}$ needs to respect:

$$|\Delta\boldsymbol{x}| \leq \Delta\boldsymbol{x}_{max} = \boldsymbol{\mathcal{K}}^{-1}\boldsymbol{\mathcal{D}}\boldsymbol{v}_{max} = 2\boldsymbol{R}\tilde{\boldsymbol{\mathcal{K}}}^{-\frac{1}{2}}\boldsymbol{R}^{T}\boldsymbol{v}_{max}, \tag{A.3}$$

obtained after using the definition of damping. Before sending to the robot, the $\Delta\boldsymbol{x}$ is *saturated* in order to respect the upper bound.

However, if taking into account the maximum static force ($\boldsymbol{F}_{max}$) when $\dot{\boldsymbol{x}} = 0$ and $\ddot{\boldsymbol{x}} = 0$, an upper bound on the stiffness can be found, such that:

$$\boldsymbol{\mathcal{K}}\Delta\boldsymbol{x}_{max} \leq \boldsymbol{F}_{max};$$

$$\boldsymbol{R}\tilde{\boldsymbol{\mathcal{K}}}\boldsymbol{R}^{T}\Delta\boldsymbol{x}_{max} \leq \boldsymbol{F}_{max};$$

$$2\tilde{\boldsymbol{\mathcal{K}}}\boldsymbol{R}^{T}\boldsymbol{R}\tilde{\boldsymbol{\mathcal{K}}}^{-1/2}\boldsymbol{R}^{T}\boldsymbol{v}_{max} \leq \boldsymbol{R}^{T}\boldsymbol{F}_{max};$$

$$2\tilde{\boldsymbol{\mathcal{K}}}^{\frac{1}{2}}\boldsymbol{R}^{T}\boldsymbol{v}_{max} \leq \boldsymbol{R}^{T}\boldsymbol{F}_{max}.$$

Hence, since the matrix $\tilde{\mathcal{K}}$ is diagonal, we can find the upper bound of each element in the i-th row and column $\left(\tilde{\mathcal{K}}_{ii}\right)$ as:

$$\tilde{\mathcal{K}}_{ii} \leq \left(\frac{(\boldsymbol{R}^T \boldsymbol{F}_{max})_i}{2(\boldsymbol{R}^T \boldsymbol{v}_{max})_i} \right)^2 \tag{A.4}$$

so, in every singular component, the value of the principal stiffness is *saturated* in order to respect the found inequality.

SAFETY IN STIFFNESS MODULATION RATE

In the context of assembly tasks, when performing variable impedance control to acquire human demonstrations or decrease the insertion force, the rate of change of stiffness cannot be too drastic. As shown in [97], a fast increase in stiffness can generate dangerous robot instability. For this reason, the stiffness modulation is done with a proportional controller with respect to the requested target one, i.e. $\dot{\mathcal{K}} = \gamma(\mathcal{K}_{target} - \mathcal{K})$.

A.2 DUAL CARTESIAN IMPEDANCE CONTROL

Differently from the execution of a single-arm, when a two-arms policy execution is performed, extra attention is required regarding the mechanical coupling of the movement. For example, when picking up a box with two hands and executing a re-shelving operation, in case of a perturbation of one arm, the other arm must also follow the perturbed movement. In this case, both arms must be *mechanically* coupled, meaning that in the impedance control of each arm, we would add an extra coupling force defined as:

$$\boldsymbol{F}_c^l = \mathcal{K}\left(\boldsymbol{x}_r - \boldsymbol{x}_l - \Delta\boldsymbol{x}_{rel}^{des}\right) + \mathcal{D}(\dot{\boldsymbol{x}}_r - \dot{\boldsymbol{x}}_l), \tag{A.5}$$

$$\boldsymbol{F}_c^r = \mathcal{K}\left(\boldsymbol{x}_l - \boldsymbol{x}_r + \Delta\boldsymbol{x}_{rel}^{des}\right) + \mathcal{D}(\dot{\boldsymbol{x}}_l - \dot{\boldsymbol{x}}_r), \tag{A.6}$$

where $\Delta\boldsymbol{x}_{rel}^{des} = \boldsymbol{x}_r^{des} - \boldsymbol{x}_l^{des}$ is the desired distance from the two end-effectors that can be learned and change over time. A simple schematic visualization of the proposed bimanual impedance control is displayed in Fig. A.1 where each end-effector is coupled with a stiffness (and damper) with respect to their goal but also with a relative stiffness (and damper) between them. Note that the proposed safety saturation and regulation process described, respectively, in Appendix A.1 are applied on a per-arm basis, thus being applied to single-arm setups. For a bimanual setup, the displacement and stiffness for the coupling forces ($\boldsymbol{F}_c$, defined in Equations (A.5) and (A.6)) are saturated and regulated similarly to Equations (A.3) and (A.4).

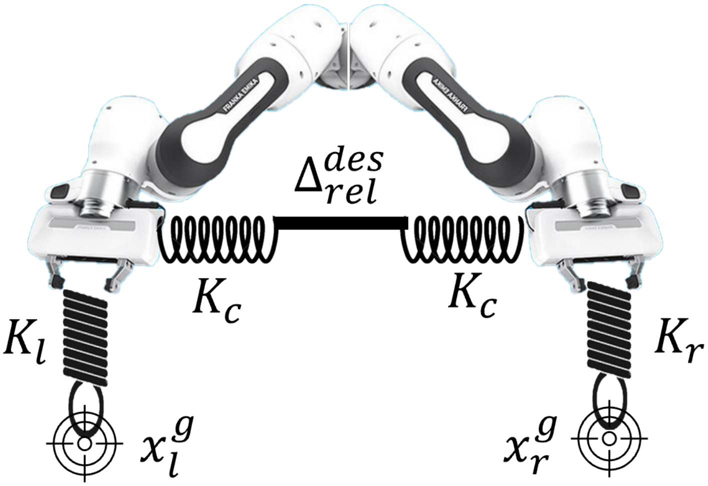


Figure A.1: The bimanual Impedance controller scheme proposed in [56]. For simplicity the spring-damper system is represented only with a spring.

A.3 EXTERNAL FORCE OBSERVER

Beyond being able to set the desired torque for each joint and simulate the effect of a Cartesian impedance, a sensor also measures the actual torque in the joints, i.e.,

$$\boldsymbol{\tau}_{sensor} = \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_{control} + \boldsymbol{\tau}_{friction}.$$

Given a force applied to the end-effector, the resulting external torque can be computed as

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}^{\top}\boldsymbol{F}_{ext},$$

thus, the external force can be estimated as:

$$\boldsymbol{F}_{ext} = \boldsymbol{J}^{\top+}\left(\boldsymbol{\tau}_{sensor} - \boldsymbol{\tau}_{friction} - \boldsymbol{\tau}_{control}\right)$$

where $\boldsymbol{\tau}_{control}$ is the final torque computed from the impedance control, and while the friction torque is obtained from a parametric estimation based on a set of parameters $C_{n,i}$, where n ranges from 1 to 4 and i ranges from 1 to 7 (corresponding to the robot's degrees of freedom) [61]. Hence, the estimated friction in each joint $\boldsymbol{\tau}_{friction}$ is given by:

$$\tau_{friction,i} = \frac{C_{1,i}}{1 + \exp(-C_{2,i}(\dot{q} + C_{3,i}))} - C_{4,i}.$$

Accurately compensating for friction is crucial when detecting external forces without an end-effector force sensor. If friction is not properly compensated, fictitious external forces may be estimated during robot motion. This can lead to issues, such as incorrect detection of kinesthetic corrections or inaccurate interpretation of discrepancies between the robot's and the human's intended movements when performing collaborative tasks [165].

References

References

[1] Fares J Abu-Dakka, Leonel Rozo, and Darwin G Caldwell. Force-based learning of variable impedance skills for robotic manipulation. In *IEEE Int. Conf. Humanoid Robot.*, 2018.

[2] Arash Ajoudani, Nikos Tsagarakis, and Antonio Bicchi. Tele-impedance: Teleoperation with impedance regulation using a body–machine interface. *Int. J. Robot. Research*, 31(13):1642–1656, 2012.

[3] Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-Based Policy Learning. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECML PKDD'11, page 12–27, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642237799.

[4] Riad Akrour, Marc Schoenauer, and Michèle Sebag. APRIL: Active Preference Learning-Based Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 116–131, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33486-3.

[5] Riad Akrour, Marc Schoenauer, Michèle Sebag, and Jean-Christophe Souplet. Programming by Feedback. In *International Conference on Machine Learning*, number 32 in JMLR Proceedings, pages 1503–1511, Pékin, China, 2014. JMLR.org. URL `https://hal.inria.fr/hal-00980839`.

[6] Alin Albu-Schaffer, Christian Ott, Udo Frese, and Gerd Hirzinger. Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*, volume 3, pages 3704–3709. IEEE, 2003.

[7] Sonya Alexandrova, Maya Cakmak, Kaijen Hsiao, and Leila Takayama. Robot programming by demonstration with interactive action visualizations. In *Robotics: Science and Systems*, 2014.

[8] Tohid Alizadeh and Batyrkhan Saduanov. Robot programming by demonstration of multiple tasks within a common environment. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 608–613. IEEE, 2017.

[9] Walid Amanhoud, Mahdi Khoramshahi, Maxime Bonnesoeur, and Aude Billard. Force adaptation in contact tasks with dynamical systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6841–6847. IEEE, 2020.

[10] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35 (4):105–120, 2014.

[11] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5):469–483, 2009.

[12] Brenna D. Argall, Eric L. Sauser, and Aude G. Billard. Tactile Guidance for Policy Adaptation. *Foundations and Trends® in Robotics*, 1(2):79–133, 2011. ISSN 1935-8253. doi: 10.1561/2300000012. URL `http://dx.doi.org/10.1561/2300000012`.

[13] Leopoldo Armesto, João Moura, Vladimir Ivan, Mustafa Suphi Erden, Antonio Sala, and Sethu Vijayakumar. Constraint-aware learning of policies by demonstration. *The International Journal of Robotics Research*, 37(13-14):1673–1689, 2018.

[14] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[15] Andrea Bajcsy, Dylan P Losey, Marcia K O'malley, and Anca D Dragan. Learning robot objectives from physical human interaction. In *Conference on Robot Learning*, pages 217–226. PMLR, 2017.

[16] Suna Bensch and Thomas Hellström. On ambiguity in robot learning from demonstration. In *International Conference on Intelligent Autonomous Systems*, 2010.

[17] Pierre Berthet-Rayne, Maura Power, Hawkeye King, and Guang-Zhong Yang. Hubot: A three state Human-Robot collaborative framework for bimanual surgical tasks based on learned models. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 715–722, 2016. doi: 10.1109/ICRA.2016.7487198.

[18] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

[19] Erdem Biyik and Dorsa Sadigh. Batch Active Preference-Based Learning of Reward Functions. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 519–528. PMLR, 2018. URL `https://proceedings.mlr.press/v87/biyik18a.html`.

[20] Andreea Bobu, Dexter RR Scobee, Jaime F Fisac, S Shankar Sastry, and Anca D Dragan. Less is more: Rethinking probabilistic models of human behavior. In *ACM/IEEE Int. Conf. Human-Robot Interaction*, 2020.

[21] Miroslav Bogdanovic, Majid Khadiv, and Ludovic Righetti. Learning variable impedance control for contact sensitive tasks. *IEEE Robot. Autom. Lett.*, 5(4):6129–6136, 2020.

[22] Wendelin Böhmer, Jost Tobias Springenberg, Joschka Boedecker, Martin Riedmiller, and Klaus Obermayer. Autonomous learning of state representations for control: An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI-Künstliche Intelligenz*, 29(4):353–362, 2015.

[23] Michael Bosongo Bombile and Aude Billard. Dual-Arm Control for Coordinated Fast Grabbing and Tossing of an Object: Proposing a New Approach. *IEEE Robotics & Automation Magazine*, pages 2–13, 2022. doi: 10.1109/MRA.2022.3177355.

[24] Cynthia Breazeal, Matt Berlin, Andrew Brooks, Jesse Gray, and Andrea L Thomaz. Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems*, 54(5):385–393, 2006.

[25] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[26] Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. *Int. J. Robot. Research*, 30(7):820–833, 2011.

[27] R. Caccavale, M. Saveriano, G. A. Fontanelli, F. Ficuciello, D. Lee, and A. Finzi. Imitation learning and attentional supervision of dual-arm structured tasks. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 66–71, 2017. doi: 10.1109/DEVLRN.2017.8329789.

[28] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9(1):1–29, 2016.

[29] Sylvain Calinon. Robot learning with task-parameterized generative models. In *Robot. Research*, pages 111–126. Springer, 2018.

[30] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2978–2983. IEEE, 2006.

[31] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[32] Sylvain Calinon, Florent D'halluin, Darwin G Caldwell, and Aude G Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 582–588. IEEE, 2009.

[33] Sylvain Calinon, Antonio Pistillo, and Darwin G. Caldwell. Encoding the time and space constraints of a task in explicit-duration Hidden Markov Model. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3413–3418, 2011. doi: 10.1109/IROS.2011.6094418.

[34] Sylvain Calinon, Zhibin Li, Tohid Alizadeh, Nikos G Tsagarakis, and Darwin G Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 323–329. IEEE, 2012.

[35] Sylvain Calinon, Petar Kormushev, and Darwin G Caldwell. Compliant skills acquisition and multi-optima policy search with em-based reinforcement learning. *Robot. Auton. Syst.*, 61(4):369–379, 2013.

[36] Gerard Canal, Guillem Alenyà, and Carme Torras. Personalization Framework for Adaptive Robotic Feeding Assistance. In *Social Robotics*, pages 22–31, Cham, 2016. Springer International Publishing. ISBN 978-3-319-47437-3.

[37] Carlos Celemin and Javier Ruiz-del Solar. An interactive framework for learning continuous actions policies based on corrective feedback. *Journal of Intelligent & Robotic Systems*, 95(1):77–97, 2019. doi: 10.1007/s10846-018-0839-z.

[38] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, Abhinav Valada, and Jens Kober. Interactive imitation learning in robotics: A survey. *Foundations and Trends® in Robotics*, 10(1-2):1–197, 2022.

[39] Stefano Chiaverini, Giuseppe Oriolo, and Anthony A Maciejewski. Redundant robots. In *Springer Handbook of Robotics*, pages 221–242. Springer, 2016.

[40] Luis Carlos Cobo, Peng Zang, Charles Lee Isbell Jr, and Andrea Lockerd Thomaz. Automatic state abstraction from demonstration. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[41] Tomás Coleman, Giovanni Franzese, and Pablo Borja. Damping design for robot manipulators. In *Human-Friendly Robotics 2022: HFR: 15th International Workshop on Human-Friendly Robotics*, pages 74–89. Springer, 2023.

[42] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30, 2017.

[43] Christian Daniel, Oliver Kroemer, Malte Viering, Jan Metz, and Jan Peters. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39:389–405, 2015.

[44] Niels Dehio, Joshua Smith, Dennis L. Wigand, Pouya Mohammadi, Michael Mistry, and Jochen J. Steil. Enabling impedance-based physical human–multi–robot collaboration: Experiments with four torque-controlled manipulators. *The International Journal of Robotics Research*, 41(1):68–84, 2022. doi: 10.1177/02783649211053650.

[45] Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Int. Conf. Machine Learning*, 2011.

[46] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[47] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems*. 2017.

[48] Marco Ewerton, Guilherme Maeda, Gerrit Kollegger, Josef Wiemeyer, and Jan Peters. Incremental imitation learning of context-dependent motor skills. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 351–358, 2016. doi: 10.1109/HUMANOIDS.2016.7803300.

[49] Yunis Fanger, Jonas Umlauft, and Sandra Hirche. Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3913–3919, 2016. doi: 10.1109/IROS.2016.7759576.

[50] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *2013 IEEE Int. Conf. Robot. Autom.*, pages 4948–4953. IEEE, 2013.

[51] Kerstin Fischer, Franziska Kirstein, Lars Christian Jensen, Norbert Krüger, Kamil Kukliński, Maria Vanessa aus der Wieschen, and Thiusius Rajeeth Savarimuthu. A comparison of types of robot control for programming by demonstration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 213–220. IEEE, 2016.

[52] Benjamin Fonooni, Thomas Hellström, and Lars-Erik Janlert. Priming as a means to reduce ambiguity in learning from demonstration. *International Journal of Social Robotics*, 8(1):5–19, 2016.

[53] Giovanni Franzese, Carlos E. Celemin, and Jens Kober. Learning interactively to resolve ambiguity in reference frame selection. In *Conf. Robot Learning*, 2020.

[54] Giovanni Franzese, Carlos Celemin, and Jens Kober. Learning Interactively to Resolve Ambiguity in Reference Frame Selection. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1298–1311. PMLR, 16–18 Nov 2021.

[55] Giovanni Franzese, Anna Mészáros, Luka Peternel, and Jens Kober. ILoSA: Interactive learning of stiffness and attractors. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7778–7785. IEEE, 2021.

[56] Giovanni Franzese, Leandro de Souza Rosa, Tim Verburg, Luka Peternel, and Jens Kober. Interactive imitation learning of bimanual movement primitives. *IEEE/ASME Transactions on Mechatronics*, pages 1–13, 2023. doi: 10.1109/TMECH.2023.3295249.

[57] Giovanni Franzese, Ravi Prakash, and Jens Kober. Generalization of task parameterized dynamical systems using gaussian process transportation. *arXiv preprint arXiv:2404.13458*, 2024.

[58] Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, 89(1-2):123–156, 2012.

[59] Andrej Gams, Bojan Nemec, Auke Jan Ijspeert, and Aleš Ude. Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks. *IEEE Transactions on Robotics*, 30(4):816–830, 2014. doi: 10.1109/TRO.2014.2304775.

[60] Manolo Garabini, Danilo Caporale, Vinicio Tincani, Alessandro Palleschi, Chiara Gabellieri, Marco Gugliotta, Alessandro Settimi, Manuel Giuseppe Catalano, Giorgio Grioli, and Lucia Pallottino. WRAPP-up: A Dual-Arm Robot for Intralogistics. *IEEE Robotics & Automation Magazine*, 28(3):50–66, 2021. doi: 10.1109/MRA.2020.3015899.

[61] Claudio Gaz, Marco Cognetti, Alexander Oliva, Paolo Robuffo Giordano, and Alessandro De Luca. Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154, 2019.

[62] Pascal Gliesche, Tobias Krick, Max Pfingsthorn, Sandra Drolshagen, Christian Kowalski, and Andreas Hein. Kinesthetic device vs. keyboard/mouse: a comparison in home care telemanipulation. *Frontiers in Robotics and AI*, 7:561015, 2020.

[63] Vinicius G. Goecks, Gregory M. Gremillion, Vernon J. Lawhern, John Valasek, and Nicholas R. Waytowich. Efficiently Combining Human Demonstrations and Interventions for Safe Training of Autonomous Systems in Real-Time. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33012462. URL `https://doi.org/10.1609/aaai.v33i01.33012462`.

[64] Andrew Gregory. Robot-assisted surgery can cut blood clot risk and speed recovery, study finds. The Guardgian, 15 May 2022.

[65] Elena Gribovskaya and Aude Billard. Combining Dynamical Systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 33–40, 2008.

[66] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L. Isbell, and Andrea Thomaz. Policy Shaping: Integrating Human Feedback with Reinforcement Learning. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2625–2633, Red Hook, NY, USA, 2013. Curran Associates Inc.

[67] Sami Haddadin, Alin Albu-Schäffer, and Gerd Hirzinger. Requirements for safe robots: Measurements, analysis and new insights. *Int. J. Robot. Research*, 28(11-12): 1507–1527, 2009.

[68] Peter A Hancock, Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, Ewart J De Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human factors*, 53(5):517–527, 2011.

[69] Sandra G Hart and Lowell E Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, volume 52, pages 139–183. Elsevier, 1988.

[70] Murtaza Hazara and Ville Kyrki. Reinforcement learning for improving imitated in-contact skills. In *IEEE Int. Conf. Humanoid Robot.*, 2016.

[71] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[72] James Hensman, Alex Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. *arXiv preprint arXiv:1411.2005*, 2014.

[73] Neville Hogan. Impedance Control: An Approach to Manipulation. In *1984 American Control Conference*, pages 304–313, 1984. doi: 10.23919/ACC.1984.4788393.

[74] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 598–608. PMLR, 2022. URL `https://proceedings.mlr.press/v164/hoque22a.html`.

[75] Jose Hoyos, Flavio Prieto, Guillem Alenyà, and Carme Torras. Incremental learning of skills in a task-parameterized gaussian mixture model. *Journal of Intelligent & Robotic Systems*, 82:81–99, 2016.

[76] Yanlong Huang, João Silvério, Leonel Rozo, and Darwin G Caldwell. Generalized task-parameterized skill learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5667–5474. IEEE, 2018.

[77] Yanlong Huang, Leonel Rozo, Jo ao Silvério, and Darwin G Caldwell. Kernelized movement primitives. *The International Journal of Robotics Research*, 38(7):833–852, 2019. doi: 10.1177/0278364919846363.

[78] Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning Trajectory Preferences for Manipulators via Iterative Improvement. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 575–583, Red Hook, NY, USA, 2013. Curran Associates Inc.

[79] Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research*, 34(10):1296–1313, 2015.

[80] Jan Verbeek. Schiphol dreigt met schrappen vluchten vanwege tekort aan bagageafhandelaars. fd.nl, 3 feb 2023.

[81] Noémie Jaquier, David Ginsbourger, and Sylvain Calinon. Learning from demonstration with model-based Gaussian process. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 247–257. PMLR, 30 Oct–01 Nov 2020.

[82] Noémie Jaquier, Leonel Rozo, Darwin G Caldwell, and Sylvain Calinon. Geometry-aware manipulability learning, tracking, and transfer. *Int. J. Robot. Research*, 2020.

[83] Charles F Jekel, Gerhard Venter, Martin P Venter, Nielen Stander, and Raphael T Haftka. Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, 12:355–378, 2019.

[84] Ravi Prakash Joshi, Nishanth Koganti, and Tomohiro Shibata. A framework for robotic clothing assistance by imitation learning. *Advanced Robotics*, 33(22):1156–1174, 2019.

[85] Mrinal Kalakrishnan, Ludovic Righetti, Peter Pastor, and Stefan Schaal. Learning force control policies for compliant manipulation. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2011.

[86] Theodora Kastritsi, Fotios Dimeas, and Zoe Doulgeri. Progressive automation with dmp synchronization and variable stiffness control. *IEEE Robot. Autom. Lett.*, 3(4): 3789–3796, 2018.

[87] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *IEEE Int. Conf. Robot. Autom.*, 2019.

[88] B. Kim, J. Park, S. Park, and S. Kang. Impedance learning for robotic contact tasks using natural actor-critic algorithm. *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, 40 (2):433–443, 2010. doi: 10.1109/TSMCB.2009.2026289.

[89] Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Trans. Robot.*, 30(5):1049–1065, 2014.

[90] W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pages 292–297. IEEE, 2008.

[91] W Bradley Knox, Cynthia Breazeal, and Peter Stone. Learning from feedback on actions past and intended. In *In Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction, Late-Breaking Reports Session (HRI 2012)*, 2012.

[92] Jens Kober, Michael Gienger, and Jochen J Steil. Learning movement primitives for force interaction tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2015.

[93] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2020.

[94] Dorothea Koert, Joni Pajarinen, Albert Schotschneider, Susanne Trick, Constantin Rothkopf, and Jan Peters. Learning intention aware online adaptation of movement primitives. *IEEE Robot. Autom. Lett.*, 4(4):3719–3726, 2019.

[95] Pallavi Koppol, Henny Admoni, and Reid Simmons. Interaction considerations in learning from humans. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[96] Oliver Kroemer, Christian Daniel, Gerhard Neumann, Herke van Hoof, and Jan Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1503–1510, 2015. doi: 10.1109/ICRA.2015.7139389.

[97] Klas Kronander and Aude Billard. Stability considerations for variable impedance control. *IEEE Trans. Robot.*, 32(5):1298–1305, 2016.

[98] Klas Kronander, Mohammad Khansari, and Aude Billard. Incremental motion learning with locally modulated dynamical systems. *Robotics and Autonomous Systems*, 70:52–62, 2015.

[99] Marco Laghi, Michele Maimeri, Mathieu Marchand, Clara Leparoux, Manuel Catalano, Arash Ajoudani, and Antonio Bicchi. Shared-Autonomy Control for Intuitive Bimanual Tele-Manipulation. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, 2018. doi: 10.1109/HUMANOIDS.2018.8625047.

[100] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[101] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 358–365, 2017. doi: 10.1109/ICRA.2017.7989046.

[102] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

[103] Quoc V Le, Alex J Smola, and Stéphane Canu. Heteroscedastic Gaussian process regression. In *22nd International Conference on Machine Learning*, pages 489–496, 2005.

[104] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

[105] Tianyu Li and Nadia Figueroa. Task transfer with stability guarantees via elastic dynamical system motion policies. In *7th Annual Conference on Robot Learning*, 2023.

[106] Nejc Likar, Bojan Nemec, Leon Žlajpah, Shingo Ando, and Aleš Ude. Adaptation of bimanual assembly tasks using iterative learning framework. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 771–776, 2015. doi: 10.1109/HUMANOIDS.2015.7363457.

[107] Robert Loftin, James MacGlashan, Bei Peng, Matthew Taylor, Michael Littman, Jeff Huang, and David Roberts. A Strategy-Aware Technique for Learning Behaviors from Discrete Human Feedback. volume 28, 2014. doi: 10.1609/aaai.v28i1.8839. URL `https://ojs.aaai.org/index.php/AAAI/article/view/8839`.

[108] Robert Loftin, Bei Peng, James MacGlashan, Michael L Littman, Matthew E Taylor, Jeff Huang, and David L Roberts. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous agents and multi-agent systems*, 30(1):30–59, 2016.

[109] Jianlan Luo, Oleg Sushkov, Rugile Pevceviciute, Wenzhao Lian, Chang Su, Mel Vecerik, Ning Ye, Stefan Schaal, and Jon Scholz. Robust Multi-Modal Policies for Industrial Assembly via Reinforcement Learning and Demonstrations: A Large-Scale Study, 2021. URL `https://arxiv.org/abs/2103.11512`.

[110] Yicheng Luo, Zhengyao Jiang, Samuel Cohen, Edward Grefenstette, and Marc Peter Deisenroth. Optimal transport for offline imitation learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[111] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, pages 50–60, 1947.

[112] Simon Manschitz, Jens Kober, Michael Gienger, and Jan Peters. Probabilistic progress prediction and sequencing of concurrent movement primitives. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 449–455, 2015. doi: 10.1109/IROS.2015.7353411.

[113] Elon Mask. Optimous folds a shirt. X (Twitter), 15 Jan 2024. URL `https://x.com/elonmusk/status/1746964887949934958?s=20`.

[114] Kunal Menda, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5041–5048, 2019. doi: 10.1109/IROS40897.2019.8968287.

[115] Cetin Mericcli, Manuela Veloso, and H Levent Akin. Multi-resolution corrective demonstration for efficient task execution and refinement. *International Journal of Social Robotics*, 4(4):423–435, 2012.

[116] Anna Mészáros, Giovanni Franzese, and Jens Kober. Learning to pick at non-zero-velocity from interactive demonstrations. *IEEE Robotics and Automation Letters*, 7 (3):6052–6059, 2022.

[117] Youssef Michel, Rahaf Rahal, Claudio Pacchierotti, Paolo Robuffo Giordano, and Dongheui Lee. Bilateral teleoperation with adaptive impedance control for contact tasks. *IEEE Robot. Autom. Lett.*, 2021.

[118] Mariano Ramirez Montero, Giovanni Franzese, Jeroen Zwanepol, and Jens Kober. Solving robot assembly tasks by combining interactive teaching and self-exploration. *arXiv preprint arXiv:2209.11530*, 2022.

[119] Toshiharu Mukai, Shinya Hirano, Hiromichi Nakashima, Yo Kato, Yuki Sakaida, Shijie Guo, and Shigeyuki Hosoe. Development of a nursing-care assistant robot RIBA that can lift a human in its arms. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5996–6001, 2010. doi: 10.1109/IROS.2010.5651735.

[120] Bojan Nemec, Nejc Likar, Andrej Gams, and Aleš Ude. Human robot cooperation with compliance adaptation along the motion trajectory. *Autonomous Robots*, 42(5): 1023–1035, 2018.

[121] Duy Nguyen-Tuong and Jan Peters. Incremental sparsification for real-time online model learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 557–564. JMLR Workshop and Conference Proceedings, 2010.

[122] Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[123] Thomas Nierhoff, Sandra Hirche, and Yoshihiko Nakamura. Spatial adaption of robot trajectories based on Laplacian trajectory editing. *Autonomous Robots*, 40:159–173, 2016.

[124] Malayandi Palan, Gleb Shevchuk, Nicholas Charles Landolfi, and Dorsa Sadigh. Learning Reward Functions by Integrating Human Demonstrations and Preferences. In *Robotics: Science and Systems*, 2019.

[125] Alexandros Paraschos, Christian Daniel, Jan Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances Neural Inf. Process. Syst.*, 2013.

[126] Rodrigo Perez-Dattari, Carlos Celemin, Giovanni Franzese, Javier Ruiz-del Solar, and Jens Kober. Interactive learning of temporal features for control: Shaping policies and state representations from human feedback. *IEEE Robot. Autom. Mag.*, 27(2): 46–54, 2020.

[127] Luka Peternel, Tadej Petrič, and Jan Babič. Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Auton. Robots*, 42 (1):1–17, 2018.

[128] Tadej Petrič, Andrej Gams, Luca Colasanto, Auke J Ijspeert, and Aleš Ude. Accelerated sensorimotor learning of compliant movement primitives. *IEEE Trans. Robot.*, 34(6):1636–1642, 2018.

[129] Emmanuel Pignat and Sylvain Calinon. Learning adaptive dressing assistance from human demonstration. *Robotics and Autonomous Systems*, 93:61–75, 2017.

[130] Emmanuel Pignat and Sylvain Calinon. Bayesian Gaussian mixture model for robotic policy imitation. *IEEE Robot. Autom. Lett.*, 4(4):4452–4458, 2019.

[131] K Pryor. Clicker training for dogs. Waltham, MA, 1999.

[132] Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Christopher KI Williams. Approximation Methods for Gaussian Process Regression. In *Large-Scale Kernel Machines*, pages 203–223. MIT Press, 2007.

[133] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *ASME J. Dynamical Syst., Measurement and Control*, 105:126–133, 1981.

[134] Daniel Rakita, Bilge Mutlu, Michael Gleicher, and Laura M. Hiatt. Shared control-based bimanual robot manipulation. *Science Robotics*, 4(30):eaaw0955, 2019. doi: 10.1126/scirobotics.aaw0955.

[135] Muhammad Asif Rana, Anqi Li, Dieter Fox, Byron Boots, Fabio Ramos, and Nathan Ratliff. Euclideanizing Flows: Diffeomorphic Reduction for Learning Stable Dynamical Systems. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 630–639. PMLR, 2020. URL https://proceedings.mlr.press/v120/rana20a.html.

[136] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 026218253X.

[137] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annu. Rev. Control Robot. Auton. Syst.*, 3:297–330, 2020.

[138] Richard Daniel and Pete Cooper. It used to be easy to get fruit pickers. BBC, 25 May 2022. URL https://www.bbc.com/news/uk-england-suffolk-61568286.

[139] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[140] Leonel Rozo, Sylvain Calinon, Darwin Caldwell, Pablo Jiménez, and Carme Torras. Learning collaborative impedance-based robot behaviors. In *AAAI Conf. Artificial Intell.*, 2013.

[141] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *IEEE International Conference on Computer Vision*, 2017.

[142] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. *Active preference-based learning of reward functions*. 2017.

[143] Seyed Sina Mirrazavi Salehian, Mahdi Khoramshahi, and Aude Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Trans. Robot.*, 32(2):462–471, 2016.

[144] Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37(10):1205–1232, 2018. doi: 10.1177/0278364918765952.

[145] Matteo Saveriano, Felix Franzel, and Dongheui Lee. Merging position and orientation motion primitives. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7041–7047. IEEE, 2019.

[146] Matteo Saveriano, Fares J Abu-Dakka, Aljaž Kramberger, and Luka Peternel. Dynamic movement primitives in robotics: A tutorial survey. *The International Journal of Robotics Research*, 42(13):1133–1184, 2023.

[147] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In *Robotics research. the eleventh international symposium*, pages 561–572. Springer, 2005.

[148] Markus Schneider and Wolfgang Ertel. Robot Learning by Demonstration with local Gaussian process regression. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 255–260, 2010. doi: 10.1109/IROS.2010.5650949.

[149] Aran Sena, Brendan Michael, and Matthew Howard. Improving task-parameterised movement learning generalisation with frame-weighted trajectory generation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4281–4287. IEEE, 2019.

[150] Burr Settles. Active learning literature survey. 2009.

[151] David Silver, J Andrew Bagnell, and Anthony Stentz. Active learning from demonstration for robust autonomous navigation. In *2012 IEEE International Conference on Robotics and Automation*, pages 200–207. IEEE, 2012.

[152] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulation–A survey. *Robotics and Autonomous Systems*, 60(10):1340–1353, 2012. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2012.07.005.

[153] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 175–184, 2004.

[154] Zheng Sun, Zhiqi Wang, Junjia Liu, Miao Li, and Fei Chen. Mixline: A Hybrid Reinforcement Learning Framework for Long-Horizon Bimanual Coffee Stirring Task. In Honghai Liu, Zhouping Yin, Lianqing Liu, Li Jiang, Guoying Gu, Xinyu Wu, and Weihong Ren, editors, *Intelligent Robotics and Applications*, pages 627–636, Cham, 2022. Springer International Publishing. ISBN 978-3-031-13844-7.

[155] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[156] AL Thomaz and C Breazeal. Adding guidance to interactive reinforcement learning. In *Proceedings of the Twentieth Conference on Artificial Intelligence (AAAI)*, 2006.

[157] Andrea L. Thomaz and Cynthia Breazeal. Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 720–725, 2007. doi: 10.1109/ROMAN.2007.4415180.

[158] Andrea Lockerd Thomaz, Cynthia Breazeal, et al. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Aaai*, volume 6, pages 1000–1005. Boston, MA, 2006.

[159] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

[160] Alexandra Topping. Almost 40% of domestic tasks could be done by robots ‘within decade’. The Guardian, 23 Feb 2023.

[161] Russell Toris, Halit Bener Suay, and Sonia Chernova. A practical comparison of three robot learning from demonstration algorithms. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 261–262. IEEE, 2012.

[162] Albert Tung, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Learning Multi-Arm Manipulation Through Collaborative Teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9212–9219, 2021. doi: 10.1109/ICRA48506.2021.9561491.

[163] Martin Tykal, Alberto Montebelli, and Ville Kyrki. Incrementally assisted kinesthetic teaching for programming by demonstration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 205–212, 2016. doi: 10.1109/HRI.2016.7451753.

[164] Peter Valletta, Rodrigo Pérez-Dattari, and Jens Kober. Imitation Learning with Inconsistent Demonstrations through Uncertainty-based Data Manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3655–3661, 2021. doi: 10.1109/ICRA48506.2021.9561686.

[165] Linda van der Spaa, Giovanni Franzese, Jens Kober, and Michael Gienger. Disagreement-Aware Variable Impedance Control for Online Learning of Physical Human-Robot Cooperation Tasks. In *ICRA 2022: IEEE International Conference on Robotcs and Automation*, 2022.

[166] Jari J van Steen, Nathan van de Wouw, and Alessandro Saccon. Robot control for simultaneous impact tasks via quadratic programming-based reference spreading. In *Proc. Amer. Control Conf.*, 2022.

[167] Steven Van Vaerenbergh, Ignacio Santamaría, Weifeng Liu, and José C Príncipe. Fixed-budget kernel recursive least-squares. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1882–1885. IEEE, 2010.

[168] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[169] Rok Vuga, Bojan Nemec, and Aleš Ude. Speed adaptation for self-improvement of skills learned from user demonstrations. *Robotica*, 34(12):2806–2822, 2016.

[170] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.

[171] Ruoshi Wen, Quentin Rouxel, Michael Mistry, Zhibin Li, and Carlo Tiseo. Collaborative Bimanual Manipulation Using Optimal Motion Adaptation and Interaction Control, 2022.

[172] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[173] Aaron Wilson, Alan Fern, and Prasad Tadepalli. A Bayesian Approach for Policy Learning from Trajectory Preference Queries. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1133–1141, Red Hook, NY, USA, 2012. Curran Associates Inc.

[174] Daan Wout, Jan Scholten, Carlos Celemin, and Jens Kober. Learning Gaussian Policies from Corrective Human Feedback. 2019. doi: 10.48550/ARXIV.1903.05216. URL `https://arxiv.org/abs/1903.05216`.

[175] Fan Xie, Alexander Chowdhury, M. Clara De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep Imitation Learning for Bimanual Robotic Manipulation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2327–2337. Curran Associates, Inc., 2020.

[176] Y. Yamada, S. Nagamatsu, and Y. Sato. Development of multi-arm robots for automobile assembly. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 2224–2229 vol.3, 1995. doi: 10.1109/ROBOT.1995.525592.

[177] Jiakai Zhang and Kyunghyun Cho. Query-Efficient Imitation Learning for End-to-End Autonomous Driving, 2016. URL `https://arxiv.org/abs/1605.06450`.

[178] Xianmin Zhang, Yanglong Zheng, Jun Ota, and Yanjiang Huang. Peg-in-Hole Assembly Based on Two-phase Scheme and F/T Sensor for Dual-arm Robot. *Sensors*, 17(9), 2017. ISSN 1424-8220. doi: 10.3390/s17092004.

[179] You Zhou, Martin Do, and Tamim Asfour. Coordinate Change Dynamic Movement Primitives – A leader-follower approach. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5481–5488, 2016. doi: 10.1109/IROS.2016.7759806.

[180] You Zhou, Martin Do, and Tamim Asfour. Learning and force adaptation for interactive actions. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1129–1134, 2016. doi: 10.1109/HUMANOIDS.2016.7803412.

[181] Jihong Zhu, Michael Gienger, and Jens Kober. Learning task-parameterized skills from few demonstrations. *IEEE Robotics and Automation Letters*, 7(2):4063–4070, 2022.

[182] Jihong Zhu, Michael Gienger, Giovanni Franzese, and Jens Kober. Do you need a hand?– a bimanual robotic dressing assistance scheme. *arXiv preprint arXiv:2301.02749*, 2023.

Glossary

BC Behavioral Cloning.

COACH COrrective Advice Communicated by Humans.

DAgger Data Aggregation.

DART Disturbances for Augmenting Robot Trajectories.

DDPGfD Deep Deterministic Policy Gradient from Demonstration.

DMP Dynamic Movement Primitive.

DoF Degree of Freedom.

DS Dynamical System.

DTW Dynamic Time Warping.

EE End-Effector.

EnsembleDAgger Ensemble Dagger.

GGP Graph Gaussian Process.

GMM Gaussian Mixture Model.

GP Gaussian Process.

GPT Gaussian Process Transportation.

HG-DAgger Human Gated DAgger.

HMM Hidden Markov Model.

IIL Interactive Imitation Learning.

IL Imitation Learning.

ILoSA Interactive Learning of Stiffness and Attractors.

Interactive RL Interactive Reinforcement Learning.

KMP Kernelized Movement Primitive.

KT kinesthetic teaching.

LfD Learning from Demonstration.

LIRA Learning Interactively to Resolve Ambiguity.

LLM Large Language Models.

LQR Linear Quadratic Regulator.

MP Movement Primitive.

MUDS Minimum Uncertainty Dynamical System.

NN Neural Network.

P&P Pick ad Place.

PI^2 Policy Improvement with Path Integrals.

PILCO Probabilistic Inference for Learning Control.

PoG Product of Gaussians.

PPL Preference-Based Policy Learning.

ProMP Probabilistic Movement Primitive.

RL Reinforcement Learning.

SafeDAgger Safe DAgger.

SHIELD Super-Human InsErtion using Learning from Demonstration.

SIMPLe Safe, Interactive Movement Primitives Learning.

TAMER Training an Agent Manually via Evaluative Reinforcement.

ThriftyDAgger Thrifty DAgger.

TP-GMM Task-Parameterized Gaussian Mixture Model.

TPC Tactile Policy Correction.

Curriculum Vitæ

Giovanni Franzese

04/10/1994	Born in San Giuseppe Vesuviano, Naples, Italy

Professional Career

09/2023-09/2024	PostDoc in Robotics Manipulation strategies for tomato harvesting Delft University of Technology Supervisor: Dr. Cosimo Della Santina

Education

28/06/2013	High School Scientific Degree Liceo Antonio Rosmini, Palma Campania, Naples
22/09/2016	Bachelor in Mechanical Engineering Politecnico Di Milano
20/12/2018	Masters of Science in Mechanical Engineering Politecnico Di Milano Track: Mechatronics and Robotics Thesis: On computing second derivatives in optimal motion planning for robot manipulators Supervisor: Dr. Alessandro Saccon (TU Enidhoven)
04/11/2024	Doctor of Philosophy in Robotics Delft University of Technology Thesis: Uncertainty-aware Interactive Imitation Learning for Robot Manipulation

Promotor: Dr. Jens Kober
Co-promotor: Dr. Luka Peternel

VISITING SCHOLAR

3/2018 - 8/2018
Master Thesis in Robotics
Eindhoven University of Technology
Supervisor: Dr. Alessandro Saccon

09/2022 - 3/2023
Visiting Research in Gaussian Process for Machine Learning
University College London
Supervisor: Prof. Marc Deisenroth

AWARDS

06/2022
Winning of Franka-Emika Manipulation Challenge (European Robotics Forum)

09/2022
TAILOR Travel Grant on Interpretable Artificial Intelligence

List of Publications

Journal Articles

🗎 Celemin, C., Pérez-Dattari, R., Chisari, E., Franzese, G., de Souza Rosa, L., Prakash, R., Ajanović, Z., Ferraz, M., Valada, A. and Kober, J., 2022. Interactive imitation learning in robotics: A survey. Foundations and Trends® in Robotics, 10(1-2), pp.1-197.

🗎 Mészáros, A., Franzese, G. and Kober, J., 2022. Learning to Pick at Non-Zero-Velocity From Interactive Demonstrations. IEEE Robotics and Automation Letters, 7(3), pp.6052-6059.

🗎 Franzese, G., de Souza Rosa, L., Verburg, T., Peternel, L. and Kober, J., 2023. Interactive imitation learning of bimanual movement primitives. IEEE/ASME Transactions on Mechatronics.

Meo, C., Franzese, G., Pezzato, C., Spahn, M. and Lanillos, P., 2022. Adaptation Through Prediction: Multisensory Active Inference Torque Control. IEEE Transactions on Cognitive and Developmental Systems, 15(1), pp.32-41.

Zhu, J., Gienger, M., Franzese, G. and Kober, J., 2023. Do You Need a Hand?–a Bimanual Robotic Dressing Assistance Scheme. IEEE Transactions on Robotics.

Conference Papers

🗎 Franzese, G., Mészáros, A., Peternel, L. and Kober, J., 2021, September. ILoSA: Interactive learning of stiffness and attractors. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 7778-7785). IEEE.

🗎 Franzese, G., Celemin, C. and Kober, J., 2020. Learning Interactively to Resolve Ambiguity in Reference Frame Selection. In CoRL (pp. 1298-1311).

Coleman, T., Franzese, G. and Borja, P., 2022, September. Damping Design for Robot Manipulators. In International Workshop on Human-Friendly Robotics (pp. 74-89). Cham: Springer International Publishing.

Ramirez Montero, M., Franzese, G., Zwanepol, J. and Kober, J., 2022. Solving Robot Assembly Tasks by Combining Interactive Teaching and Self-Exploration. arXiv preprint arXiv:2209.11530.

Bootsma, B., Franzese, G. and Kober, J., 2021, August. Interactive learning of sensor policy fusion. In 2021 30th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (pp. 665-670). IEEE.

Cosier, L., Iordan, R., Zwane, S., Franzese, G., Wilson, J.T., Deisenroth, M.P., Terenin, A. and Bekiroglu, Y., 2023. A Unifying Variational Framework for Gaussian Process Motion Planning. arXiv preprint arXiv:2309.00854.

PRE-PRINTS

- 🗎 Franzese, G., Prakash, R., Kober, J., 2024. Generalization of Task Parameterized Dynamical Systems using Gaussian Process Transportation

WORKSHOP PAPERS

van der Spaa, L., Franzese, G., Kober, J. and Gienger, M., 2022. Disagreement-aware variable impedance control for online learning of physical human-robot cooperation tasks. In ICRA 2022: IEEE International Conference on Robotcs and Automation.

AWARDS

- 🏆 Best Late Breaking Results Poster Award in 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2021) for ILoSA: Interactive learning of stiffness and attractors.

🗎 Included in this thesis.
🏆 Won a best paper, tool demonstration, or proposal award.

Acknowledgments

Thank you for reaching the final pages of this manuscript. Many curious readers may not have gone through all the details. I don't blame you, and I hope you can enjoy at least some of the figures as much as I enjoyed creating them. I thoroughly enjoyed my entire PhD journey in Delft, from the interview process to these final days, including navigating a global pandemic and an exciting exchange in London. These last pages are to thank the people who made this book possible.

First, I would like to thank my PhD committee—Robert Babuska, Manon Kok, Bojan Nemec, Alessandro Saccon, and David Abbink—who took on the challenging task of reading, evaluating, and providing valuable feedback on this manuscript.

However, my PhD journey would not have been possible without my advisor, Jens Kober, who selected me from a large pool of candidates and inspired and mentored me throughout the four years of graduate school. Starting a PhD in robotics in Delft was a dream come true for me. Thank you, Jens, for always encouraging me (often implicitly) to do better while allowing me to explore new ideas. You never imposed your perspective on research, which made me feel like a true scientist. Thank you for letting me make my own mistakes, which you pointed out with kindness, and for celebrating my discoveries with humility. I am also grateful to you for fostering an amazing interdisciplinary research group where we all had the opportunity to learn from one another at every moment.

Nevertheless, I was lucky to have a co-promotor, Luka Peternel. From February 2021, I had the pleasure of being advised by him weekly, and our long conversations brought many ideas. Thank you, Luka, for being a passionate roboticist and advisor and helping me in the discovery and design process of the algorithms of this thesis. Thank you for supporting me when recovering from rejections and other challenging times with your philosophy pills.

A special thanks also to Marc Deisenroth for welcoming me to UCL and mentoring me during my visit there. Working on Gaussian Process in his group had a different taste, and I am glad I could take a bite.

Also, many thanks to Cosimo Della Santina, who advised me in my last year in Delft during my “PostDoc". Thank you, Cosimo, for trusting me to teach the Gaussian Process lecture in your course for three consecutive years. Teaching one of my favorite topics to young engineers has been one of my best academic career experiences.

My PhD dissertation is not solely the result of my efforts and those of my supervisors but a product of numerous collaborations and cross-disciplinary exchanges within my department and beyond. Among my co-authors are PhD peers, Master's students, and Postdocs, without whom this thesis would not have been possible. I want to thank Rodrigo and Carlos for introducing me to interactive learning and giving me a warm introduction to the field. Your enthusiasm and passion pushed me to test the limits of robotics and explore new ideas continually. Rodrigo, in particular, became my housemate. He somehow tolerated my bad jokes, even after spending entire days together in that small house by the canal. Our dinner conversations about science positively impacted my PhD creativity. Every PhD student deserves a housemate like Rodrigo, but I was the lucky one.

Various Master's students and their theses intersected with my winding PhD journey. Being surrounded by them made the long road more enjoyable, filled with constructive discussions, robot debugging sessions, and paper writing. I especially want to thank Anna, Tim, and Bart, who supported me in exploring ideas ranging from teaching a robot how to grasp objects quickly to performing bimanual manipulation to autonomous driving—all through human demonstrations.

The European Project TERI funded my PhD project, and as an EU citizen, I feel both grateful and proud to have had this opportunity. I would like to thank Leandro, Zlatan, Ravi, and Marta for joining me in the challenging process of enabling robots to learn interactively from humans. Many thanks also to Jihong, who proved how interactive learning could be practical for teaching robots how to dress humans and involved me in a futuristic (but necessary) research topic.

My PhD journey was filled with robot experiments. Still, the most remarkable adventure was deploying the ideas from this thesis to tackle the Robothon challenge in 2023 alongside the Platonics team. Applying research ideas to a real-world manipulation problem opened my eyes to many challenges we often overlook in lab experiments. Thank you, Mariano, Max, and Chadi, for striving to achieve great things together and creating some of my most memorable moments in Delft.

I would also like to express my gratitude to Corrado and Mert, as well as the entire AirLab team, for making the days spent in RoboHouse fun and for helping me decode the many strange behaviors of our Panda robots.

Coming from abroad and starting anew is never easy. I am grateful to all the colleagues who became close friends. In particular, I want to thank Alvaro for our endless conversations about life and learning; Linda, who welcomed me into the department and office and even tried to teach me to sing—with no success; Bruno and Hai for inspiring and motivating the young PhD students to achieve great things and for creating a lasting vision for the department; Tasos for our Socratic walks and philosophical discussions on ethics, fairness, and science; Lasse for always sharing his opinions and insights during our lunchtime discussions; Tomas for teaming up with me to tame many master students and avoid irreversible damages for the lab; Bas and Jelle for proudly representing and defending Dutch culture in a very international department; Elia, for proving that you don't need to be born near Mount Vesuvius to make great pizza.

To my London friends. To Marco Piva, who welcomed me as an old friend and was my main lighthouse to navigate the new city. To Mari and Sunshine, my housemates, who became like two sisters to me in a very short time. Our movie nights and dinners (with Laurens) on a broken table are safely stored in the casket of good memories.

To the Statistical Machine Learning lab, 'Kwanda, Yasemin, Vignesh, So, Jake, Daniel, Mathieu, and Denis. Thank you for being very friendly and giving me the warmest welcome and best possible stay in London.

Earning a PhD involves navigating many challenging times, and my family, partner, and friends have been there to support me countless times. I would like to thank my friends Lorenzo, Italo, and Manuel for always cheering me up when I was upset and adding a touch of irony to my prickly tendencies. You made many days in the office and evenings genuinely memorable. Thank you, guys, for bringing color to many grey weeks, and I am sorry for all the times I fell asleep on the sofa while we were watching a movie.

To my Politecnico friends: Fois, Diego, Ivagnes, Leo, Asso, Notario, Nich, Fabio, and Giulio. Facing the challenging years at PoliMi together created a unique bond that will never dissolve, even though we are now scattered around Europe and no longer share the same classrooms. I hope one day we will all finally live in the same city and go on trips on the weekend with our own “ciamioncino".

To my mum Geni, dad Michele, and brother Vittorio for being the best family on the planet. Growing up with you sharpened my critical thinking and continuously allowed me to refine my ideas. It has been 11 years since I left our home in Palma Campania, missing many family gatherings and episodes of daily life. This hasn’t been easy, neither for me nor for you, and I would like to thank you for always supporting me in following my dreams, even when they imply being far away for a long time. This thesis is also yours.

To my lovely girlfriend Beatriz. You stood by me and supported me, giving me more time than you received in return without making me feel obligated. I am forever grateful for the care you provided and the love, culture, and countless books you brought into our home. I promise that I will be more careful during the house chores or design a robot that can do that for us.

In reality, I do not know if robots will be able to take on the tasks I prefer to avoid, but I hope they will not replace what I genuinely enjoy, like generating new knowledge.

Giovanni
Delft, September 2024

Propositions

accompanying the dissertation

UNCERTAINTY-AWARE INTERACTIVE IMITATION LEARNING FOR ROBOT MANIPULATION

by

Giovanni FRANZESE

1. Machines memorize. They do not learn.
2. Every scientific breakthrough in robotics, whether in software development or the realm of knowledge, should be made accessible to the public through open access.
3. In robotics user studies, a careful review process is essential to prevent biased or misleading research from being published.
4. Gaussian Process is a clearer and more transparent path toward interpretable and reliable models compared to Deep Learning equivalents. [Chapter 7]
5. The most effective and efficient approach to teaching a complex policy does not solely rely on demonstrations but rather leverages feedback from human interactions. [This thesis]
6. Utilizing imitation learning combined with classical control offers a safer and more efficient method for learning and executing complex manipulation tasks compared to lengthy reinforcement learning-based training. [This thesis]
7. Academics should prioritize teaching quality over prolific yet incomplete publications, as this emphasis, paradoxically, enhances both education and the quality of research outcomes.
8. The "TikTok-ification" of education—cramming 20 topics into a 1-hour lecture—makes future scientists more superficial and less able to focus on complex topics later in their careers.
9. There are no valid reasons to dislike ROS (Robot Operating System).
10. Manipulators must be used to manipulate, nothing else.